Analyzing and Repairing Concept Drift Adaptation in Data Stream Classification

Anonymous

***Abstract*—Pollution from wood burners has serious health implications for residents of rural towns, even in developed countries. Monitoring the level of airborne particulate matter, $PM_{2.5}$, often requires making inferences about missing or corrupted readings. Air Quality inference in these cases poses two key challenges common to many data stream classification tasks. Firstly, it displays non-linear spatio-temporal relationships dependent on many factors, *e.g.* weather and traffic. Secondly, these factors can evolve over time, changing the distribution of data. For example, changing wind direction can have a large impact on which neighboring sensors are most influential during inference. Change may be able to be captured by incorporating relevant factors into the model, however in many situation we have access to few if any of these factors. In these cases, alternate approaches must be taken to detecting and adapting to change not present in available training data. We propose a data stream based classification system, called AirStream, that is able to detect and adapt to change in unknown environmental factors. Such changes in the distribution of data are known as *concept drift*. We show that by treating the data set as a stream and learning incrementally, concept drift detection methods can allow systems to react to unseen change in their environment. Using the air quality inference task, we analyse the relationship between adaption to concept drift and change in environmental factors. We discovered a strong predictive link between the adaptions made by AirStream and changes in meteorological conditions. Supported by a novel repairing algorithm to identify and correct errors in concept drift adaption, we found AirStream provided gains in classification performance compared to seven baseline methods.**

***Index Terms*—Concept Drift, Data Stream classification, Air Quality**

I. Introduction

Air pollution contributes to nearly 9 million deaths worldwide every year [1], most of which are related to the use of wood-fired heaters and cookers. The use of wood as a fuel in domestic appliances is a significant source of $PM_{2.5}$ (particles smaller than 2.5 μm in diameter) and many other dangerous pollutants. Government monitoring agencies are increasingly seeking detailed air quality data to inform residents and detect conditions associated with high $PM_{2.5}$ levels. Given the cost of air quality instruments, this monitoring has traditionally taken the form of a single measurement point taken as representative of the whole urban area. Over the past few years, monitoring has shifted towards multiple low-cost air quality sensors distributed across an area. However, these low-cost sensors often suffer from missing data, calibration drift and unstable operation. This leads to the presence of missing or corrupted readings that need to be inferred so that the whole data set retains its usefulness. An example of the inference problem we consider here is shown in Figure 1. The current level of $PM_{2.5}$ at a target location T must be predicted using the readings of neighboring sensors at times t and $t-1$, and the level of T at $t-1$ if it is available.

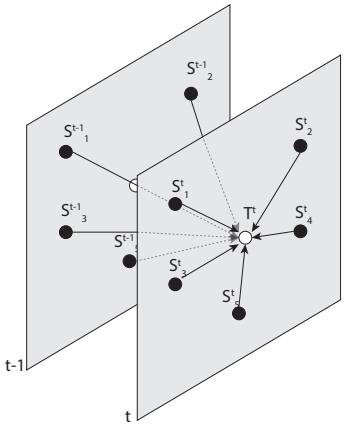

Fig. 1: An example of inference. The level of $PM_{2.5}$ at location T at time t is predicted using the values of neighboring sensors at times t and $t-1$.

Inference in air quality data is difficult due to spatial non-linearities and abrupt temporal changes. These often lead to the readings from two geographically close sensors, or temporally close readings from a single sensor, being very different. Past research [2], [3] has identified strong dependencies between these spatio-temporal relationships and environmental and contextual features, such as, meteorological conditions (wind), urban activity (traffic and heater use) and points of interest. For example, changing wind direction might change the direction in which pollution flows between sensors (illustrated by Figure 2), or falling temperatures might encourage the use of wood burners, thus, increasing the proportion of pollution from residential areas. These changes influence which sensor readings are most informative when making inferences. In other words, environmental conditions may impact the inference relationship between *features* and *label*. This problem is not limited to the air quality domain, and is commonly known as *concept drift* [4]. Classifiers without the ability to detect and adapt to changes in conditions have difficulty retaining performance as conditions change over long periods of time, for example the 12 week studies we investigate here. Figure 3 shows that the performance of a linear inverse distance weighted interpolator (IDW) [5] and an Ordinary Kriging classifier is unstable as wind direction changes across a longer period of time.

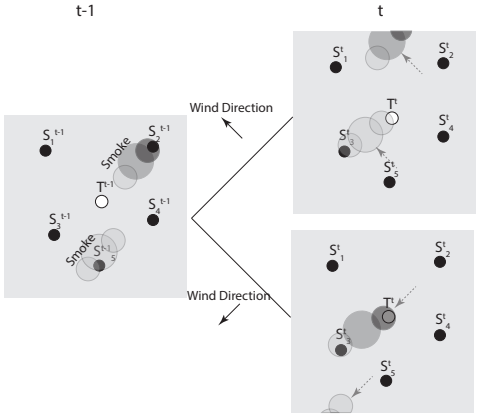

Fig. 2: An example of different spatio-temporal relationships due to wind. In the top right figure, wind from the southeast blows smoke from S_5 to T so $T^t = S_5^{t-1}$. In the bottom right figure, wind from the northeast blows smoke from S_2 to T so $T^t = S_2^{t-1}$.

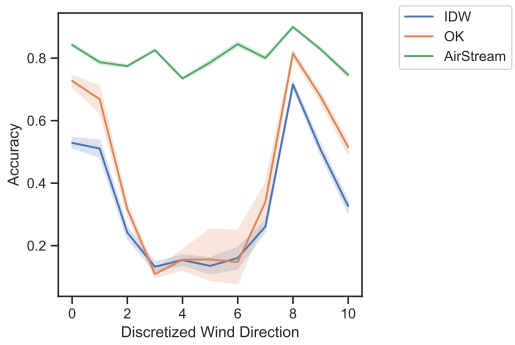

Fig. 3: Performance of linear interpolation (IDW), Ordinary Kriging (OK) and AirStream classifiers as wind direction changes.

A common solution to concept drift is to incorporate relevant environmental features into the model to better capture change. Dispersion models have been used to simulate the transport and transformation of pollutants across an area. These models are generally mechanistic rather than heuristic and their performance is dependent on the quality of information available about large numbers of parameters relating to the current environment. Other methods learn the effects of hand selected environment features on the inference relationship. For example, Cheng et al. [6] used weather type, temperature, pressure, humidity, wind, points of interest and traffic features in an attention based neural network to determine which neighboring sensors will be most influential at each prediction.

This solution is not always practical due to the large amount of data and data sources it requires. A common challenge is a lack of good quality information for the environmental features relevant to concept drift. In the two locations investigated in this research, one has only wind speed and direction available while the other has no reliable meteorological monitoring at all. Inference systems which require contextual features are unusable or suffer poor performance in these areas.

The air quality inference problem exemplifies the need for a robust classification system capable of adapting to concept drift without relying on additional environmental data. We propose AirStream to solve this problem, a general data stream framework capable of detecting and adapting to change even in unobserved features. When applied to the air quality inference task, AirStream provides high accuracy air quality level inference using only sparse spatial and temporal readings from neighboring sensors, allowing it to be used in locations lacking rich feature sets. We show, using real data taken in two rural towns, that our system is capable of identifying changes related to wind direction and speed without those variables being available as input.

We consider observations as a stream, allowing the application of data stream mining techniques. Rather than adapting to environmental features, we adapt to changes in the distribution of streaming observations using concept drift detection methods. Reacting to these changes lets us consider the stream as a sequence of stationary segments, allowing the application of powerful stationary classifiers. As each new segment is encountered, we build a new classifier for it or reuse an old classifier. AirStream builds on top of this framework by introducing a repair algorithm to increase to robustness of this adaption process, allowing us to track transitions between classifiers to build a model of changing conditions. As AirStream works directly with the raw readings of potentially unreliable sensors, we expect the introduction of noise. This noise has the potential to disrupt the transition process by masking a change in conditions or hiding the correct classifier for a segment. The repair algorithm allows AirStream to detect and repair these potential errors, allowing the reuse of classifiers in the system to be a strong predictor of recurring environmental conditions. Crucially, we apply this model of changing conditions to infer meteorological information directly from the classification process.

We deployed our system on data measuring the air quality of two rural towns, Rangiora and Arrowtown. AirStream obtained a higher predictive performance in inferring $PM_{2.5}$ levels across all target locations compared to seven other baseline methods. The changes captured by our system are shown to increase the ability to infer current environmental conditions above only air quality features.

We present the following contributions in this paper:

- We propose a data stream based system, AirStream, capable of adapting to changing levels in unknown environmental conditions. This allows the system to be used in situations where state-of-the-art competitors cannot due to lack of data.
- We propose a method of detecting and repairing concept drift adaption errors. By periodically sampling the accuracy of inactive classifiers, we identify cases where change was missed or misclassified. Repairing these errors increases performance and produces a more robust

model of changing conditions.

- We perform an analysis of our condition model, verifying the changes we detect are linked to changes in weather conditions. We investigate the ability to use the changes we detect in inference of current environmental conditions.

In the next section we present an overview of the inference problem and discuss the data sets we investigate. In Section III we outline AirStream and in Section IV we discuss a component to repair decision making errors due to noise. In Section V we overview the basis for inferring environmental conditions using AirStream. We evaluate AirStream's ability to infer both $PM_{2.5}$ and environmental conditions in Section VI.

II. Problem Overview

In this section we briefly introduce the data sets investigated in this work and discuss the challenges they pose. We also introduce data stream mining methods to handle these challenges.

A. Rural AQI Data

To combat high rates of wood smoke pollution in rural areas, two government run studies placed ODIN wood smoke pollution sensors around rural towns. The first study, *Rangiora*, placed 13 sensors over winter, from 20 June 2017 to 25 August 2017. The second study, *Arrowtown*, placed 51 sensors over the three months between 16 July 2019 and 18 September 2019. Each sensor produced 3 readings, PM_1, $PM_{2.5}$ and PM_{10}, in minute intervals. We select $PM_{2.5}$ as the target for this work. Huggard et al. [7] identifies $PM_{2.5}$ as having the most important impact on human health, recording the level of particulates small enough to harm human lungs, while also being the most accurately measured of the ODIN particulate matter readings.

Some sensors were only activated partway through the period, and many suffered breakages or missed readings. In order to accurately assess performance against ground truth, we select a subset of each data set with all sensors active. For *Rangiora* we select a segment of 9 sensors across 53,810 observations. For *Arrowtown* we select 10 sensors across 68,000 observations.

Each data point in the raw data consists of a timestamp, the serial number of the sending sensor and numeric readings for each of the 3 measures. To preprocess the data set, we first align the timing of readings by rounding to the nearest minute. One sensor is designated a target, and its readings are discretized into 6 levels according to international $PM_{2.5}$ quality recommendations [8]. The $PM_{2.5}$ breakpoints of these levels are shown in Figure I. The locations of sensors in *Rangiora* and *Arrowtown* and the distribution of observed $PM_{2.5}$ levels are shown in Figure 4. We also show the locations and distribution of a similar benchmark data set from Beijing [9]. We evaluate AirStream on these data sets in Section VI.

The classification task is to predict the target AQI level based on the current and previous readings of all non-target

TABLE I: $PM_{2.5}$ Levels (Based on 24 Hour Averages)

Level	Category	Low	High
0	Good	0.0	12.0
1	Moderate	12.1	35.4
2	Unhealthy for sensitive groups	35.5	55.4
3	Unhealth	55.5	150.4
4	Very Unhealth	150.5	250.4
5	Hazardous	250.5	¿250.5

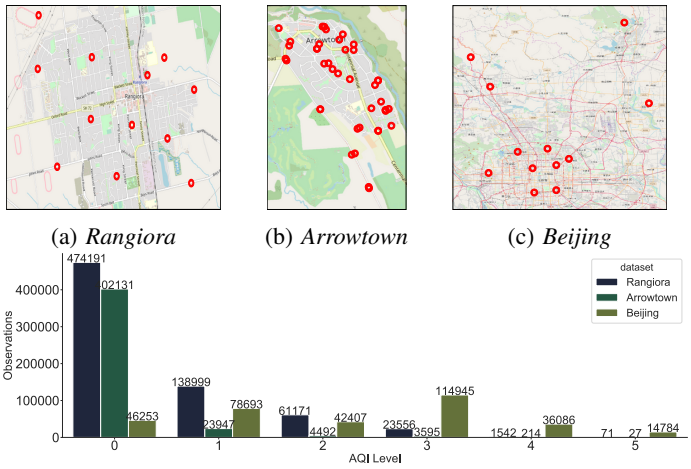

Fig. 4: Sensor locations and distribution of $PM_{2.5}$ levels

sensors and last stable reading of the target sensor. Formally, if the sequence of readings for the sensor with index i is $S_i = (s_i^1, s_i^2, \ldots, s_i^t)$, at time t we attempt to classify the level of the target sensor l, s_l^t with features $\{s_i^t, s_i^{t-1} | i \neq l\} \cup \{s_l^{t_l}\}$, where $s_l^{t_l}$ is the most recent reading received from the target sensor. In this paper we investigate interpolation, however the task could easily be adapted to the prediction of a timestep f steps in the future by inserting s_l^t into the feature set and instead classifiying s_l^{t+f}.

Challenges. The goal of this classification task is to improve the quality of air quality measurement by inferring missing or unreliable readings. A secondary goal is to provide information on current environmental conditions so they may be linked to periods of poor air quality.

An important distinction between this task and past work is the lack of rich multi-source environmental features. These often include features such as wind speed, temperature or pollution sources like traffic density. Air pollution has complex non-linear spatial and temporal relationships which are dependent on these features. The classification task here is to not only work without these features, but to provide some signal towards what these features may be. In Section VII we discuss Neural Network based methods which have achieved strong performance in similar tasks where large training sets and monitored environmental features are available, however these methods do not detect and adapt to previously unseen changes in factors not incorporated into the model thus are unsuitable for this particular task. Additionally, these methods do not allow us to analyse the sequence of adaptions made in response

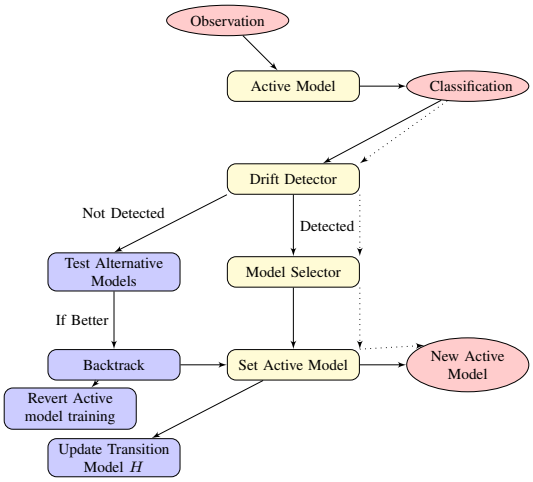

Fig. 5: AirStream System Overview. Dotted lines indicate a basic data stream framework flow, solid lines indicate AirStreams flow, including our additional components in blue.

to changing conditions. The data stream approach proposed in the next section allows us to identify, model and adapt to changes in environmental conditions without environmental features.

B. Data Stream Overview

We can consider a data set as a stream of observations. Changes in environmental conditions can change the relationship between features and labels as the stream progresses. For example, a change in wind direction may shift which sensors are upwind of the target location, changing spatial relationships important to the classification task. Such a change in the feature to label relationship is known as *concept drift*. If concept drift is not dealt with correctly, it may lead to poor classifier performance.

A data stream framework provides tools which allow us to consider a potentially non-stationary data set containing concept drift instead as a sequence of stationary segments. This approach allows the application of powerful stationary classifiers in non-stationary environments. By learning incrementally, a system in a streaming setting can safely assume the arriving data is from a similar distribution until a concept drift is detected. A crucial characteristic provided by this approach, and required for this specific problem, is the ability to detect changes in unknown (hidden) features not observed by the modeling process. Once a drift is detected, an adaption process can then be triggered to react to the drift.

The next section describes how we apply these data stream classification methods to real-time inference of air quality levels. In addition to this basic framework, we propose a technique to handle noise introduced into the system by unreliable sensors.

III. Inference System

We first give an overview of our AirStream system, shown in Figure 5 before describing in detail our proposed components.

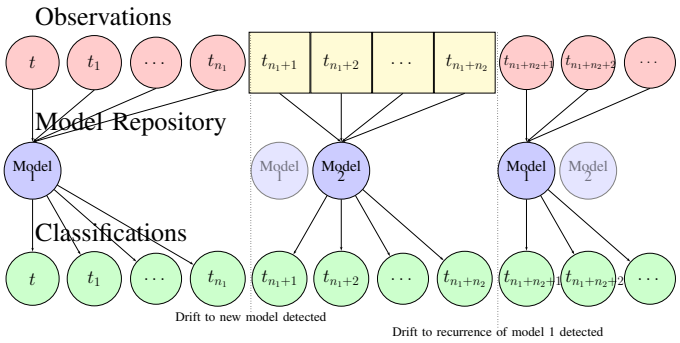

Fig. 6: Basic data stream framework example timeline showing a transition to a new model and transition to a recurring model.

We will briefly introduce two important components of data stream classification, incremental learners and drift detectors. An incrementally learning classifier incorporates observations sequentially. Unlike a batch trained model, an incremental learner can make predictions anytime with no training downtime. This allows rapid adaption to change by creating a new classifier. Typically there is a 'recovery period' while the error rate drops to a stable level as the new classifier trains. The incremental learner we use here is the Hoeffding Tree [10], an online decision tree method used in state of the art methods [11]. The Hoeffding Tree was chosen as it provides strong performance guarantees, and provides a representation of a concept that can only grow. This provides a more robust representation of a concept, allowing tracking over the course of the stream.

We detect change using a drift detector. In a streaming setting, observations are processed sequentially. By considering the error rate of a stationary classifier, we can detect concept drift as a statistically significant change in error rate. We use the ADWIN drift detector [12], a dynamic windowing based technique.

These two components are the basis of the data stream framework shown in Figures 6 and 7. A Hoeffding tree is initialized as the *active model*. The active model is incrementally trained on incoming observations with its error rate fed into the drift detector. When a drift is signalled by the detector, the active model is deactivated and added to a set of inactive models. A *model selection process* is initiated to select a new classifier best suited to the new stationary stream segment. The model selection process compares the performance of all inactive models, as well as a newly built model, over a recent window. This process attempts to pick the model most suitable for the emerging segment. The best performing model is selected as the new active model. This process repeats for each new segment in the stream. Figure 6 shows an example stream with three segments. A drift is signalled at time t_{n_1}, and the active model transitions to a fresh model, Model 2. Model 2 is used to classify observations until the next concept drift is detected at t_{n_2}. The model selection process matches the original Model 1 as the most suitable model.

We propose additions to this basic framework, shown in

Fig. 7: Proposed system example timeline, including the repair of a false positive error and a false negative error.

blue in Figure 5. Intuitively, if we chose to reuse the same model on a different segment of the stream this indicates that the segments displayed a similar relationship between features and label. We hypothesize that a similar relationship between features and label indicates similar environmental conditions. A model can then be considered as a representation of some set of environmental conditions. By building and maintaining a history of active models, we can infer environmental changes based on the transitions made during this history. We define the *Model History*, M_H, of our system as the sequence of active models used to classify the observations in a data set.

$$M_H = (A^1, A^2, ..., A^t),$$

where A^t is the ID of the active model used to classify the observation at time t.

We treat the air quality readings taken as input by AirStream as potential sources of noise. To keep models as clean representations of environmental conditions given the noisy readings generated by unreliable sensors, we implement a method to repair sub-optimal transitions. We detail this component in the following section.

IV. Drift Repair

AirStream attempts to use the model which best represents the relationship between features and label to classify each observation. Noise can lead to a sub-optimal model being selected, *i.e.*, where selecting an alternative inactive model or building a new model would have better represented the relationship displayed by incoming observations.

We observe two types of such selection errors, which we refer to as False Positive and False Negative. False Positive errors occur when the model selection process selects a sub-optimal classifier to transition to. This is shown in Figure 7 at point (II). As model selection is based on performance over a window of observations, noise in this window can make it difficult to determine the most suitable model to use. False negative errors occur when a drift is not detected, and the system remains using a sub-optimal model rather than transitioning to the true model. This can occur when fluctuations in error rate due to noise mask an error rate change due to concept drift. This is shown in Figure 7 at point (IV).

These errors have three main negative effects. Firstly, error rate is increased due to the use of a classifier miss-aligned with the true feature-label relationship. Secondly, the sub-optimal model's representation of its initial relationship is also degraded as it is trained on observations displaying a different relationship. Lastly, missing and miss-labeled transitions between models are less associated with changes in environmental conditions.

We propose a drift repair mechanism to mitigate these cases. We identify false positive and false negative errors by periodically testing inactive models against the active model. A sustained performance above the active model or a sharp change in relative performance indicates the tested inactive model may be a better fit for the current stream and a False Positive or Negative error may have occurred. If either of these indicators pass a confidence threshold, we restore the system to the point where testing started in a *backtrack* repair. While we cannot reverse the effects of classifications made during this time period, we are able to reverse any model training carried out and repair our transition model. This ensures models do not incorporate training information from the wrong concept, and keeps meta-information clean. The *backtrack* process is described at the end of this section.

Formally, we place a *restore point* after each transition and periodically every R_p observations, creating a backup of the active model. We then select the K best performing inactive models on a recent window as alternative models to test. For the next R_l observations we calculate the running Kappa statistic κ_a for the current active model, a, and each of the alternatives $k \in K$, κ_k. A confidence threshold is calculated to determine if any alternative models are a better fit for the current stream. This confidence threshold is made up of two parts. To capture gradual performance change we implement a statistical confidence test and to capture quick change we implement a change detection scheme. Gradual change confidence C_g is determined by a one-sided t-test at each observation testing the null hypothesis that κ_k is less than or equal to κ_a. If the average p-value over the testing period falls below a threshold, *i.e.* 0.05, we determine model k is a better fit than the active model and a *backtrack* repair is initiated. To capture quick changes in model performance, we run the ADWIN change detection method on the performance difference $\kappa_k - \kappa_a$. A detection of change in this stream while $\kappa_k - \kappa_a$ is positive and increasing determines model k has become a better fit than the active model and a *backtrack* repair is initiated. This quick change could come from a change in distribution missed by concept drift detection.

When a *backtrack* repair is initiated, we deactivate the active model and revert it to its state at the latest restore point. If the latest restore point was at a transition we classify the error as a False Positive error, *i.e.*, the transition was to the wrong model. If the latest restore point was not at a transition, we classify the error as a False Negative error, *i.e.*, a drift occurred and was not detected, leaving the system using a sub-optimal model. In both cases model k is activated and the system continues in its new state.

V. Condition Inference

When AirStream detects a concept drift, the active model is changed to one that matches recent observations. We hypothesize that a major driver of these concept drifts is change in environmental conditions. This hypothesis indicates that the transitions between active models in AirStream are linked to changes in environmental conditions, and further, that when a model is reused similar environmental conditions

are present. Under this hypothesis, matching weather conditions to AirStream active models may allow conditions to be inferred in the future. For example, consider a scenario where meteorological data was temporarily recorded in a location where AirStream was active. If the post-hoc analysis revealed a strong relationship between a given set of weather conditions and the use of a given model, we may infer the similar weather conditions are present the next time the active model is used. We consider two methods for relating a set of environmental conditions to the use of an active model, a recall and precision based method and a classifier based method.

We denote the *Condition History* C_e as the sequence of discretized observations of a given environmental source e over the time period of a particular data set, $C_e = (E^1, E^2, ..., E^t)$, where E^t is the level of e at time t.

We evaluate the relationship between model use and environmental conditions by matching patterns in M_H and C_e. We consider the precision, $P(m, l)$, and recall, $R(m, l)$, obtained by matching each environment level, l to the use of a particular active model m.

$$R(m,l) = \frac{|\{t|A^t = m, E^t = l\}|}{|\{t|E^t = l\}|}$$

$$P(m,l) = \frac{|\{\{t|A^t = m, E^t = l\}|}{|\{t|A^t = m\}|}$$

$$F1_c(m,l) = 2\frac{R(m,l)P(m,l)}{R(m,l) + P(m,l)}.$$

In this case, precision measures the strength of the relationship 'model m is active therefore e has the value l' while recall measures the proportion of observations where $e = l$ where this relationship holds. The $F1_c$ score combining recall and precision measures the overall strength of the relationship.

We also consider the ability to train a secondary machine learning model to predict the level of e given the current active model. We train a classifier using M_H as the set of features and C_e as the set of labels. The predictive ability of this classifier indicates the strength of the relationship between M_h and C_e.

We investigate the strength of the relationship between the active model used by AirStream and environmental conditions in the next section, and find evidence that both wind speed and direction can be inferred through these methods in the *Rangiora* data set.

VI. Evaluations

In this section we evaluate the accuracy of the $PM_{2.5}$ levels inferred by our proposed system. We first describe how AirStream was applied to infer $PM_{2.5}$ levels in *Rangiora* and *Arrowtown* for the New Zealand National Institute of Water and Atmospheric Research (NIWA), comparing to seven baselines ranging from interpolation methods to a state-of-the-art data stream algorithm. We also describe an experiment applying the system to a benchmark data set of urban $PM_{2.5}$ readings taken in Beijing, China. We then investigate the link between changes detected by our system and changes in environmental conditions.

Experimental Setup. We apply AirStream to infer $PM_{2.5}$ levels in two rural towns given the presence of missing labels, as described in Section II-A. We also evaluate on a similar data set of readings taken in Beijing [9]. There are three main differences in this data set compared to the other data sets. Firstly it is an urban environment compared to small rural towns. Secondly the sensors are spread much further apart (across the Beijing metro area), and lastly all readings are recorded every hour compared to every minute. For all data sets, at each observation a classifier has access to the current and previous numeric readings of surrounding sensors, as well as the last seen $PM_{2.5}$ level of a target sensor. The task is to infer the current $PM_{2.5}$ level of the target sensor.

To allow evaluation against ground truth, we select a portion of the data set where all labels are available and randomly mask labels to simulate missing sensor readings. We select b observations as 'broken', and mask these labels. Most sensor breakages in the data set last for longer than one minute, thus appear as blocks of missing readings rather than single observations. To recreate this effect we select the b_{period} sensors following the initial b breakages to also be masked. When a label is masked it is not available in the feature set of the next observation, and cannot be trained on. We set b as 3% of the size of each data set with $b_{period} = 60$. All experiments are repeated 10 times on all possible target sensors with the average results being reported.

We also ran AirStream on synthetic data sets created using the TREE and RBF generators available in MOA [13].

We compare our system to seven baseline methods in four categories: simple naïve methods, spatial interpolation methods, spatial and temporal methods and competing data stream methods.

1) *Chance and No-Change Classifiers (NC)*: These simple methods predict a random label drawn from the distribution of a given classifier, and most recent non-masked label, respectively. The No-Change classifier has been shown to be very effective in many data stream classification tasks affected by auto-correlation [14]. We compare to the chance classifier by using a Kappa Statistic measure [14] in our evaluation.
2) *Inverse Distance Weighted interpolation (IDW) and Gaussian interpolation*: These methods infer the current target $PM_{2.5}$ level by interpolating current readings from surrounding sensors. IDW uses the average reading weighted by inverse distance to the target, while Gaussian interpolation assumes pollution reported by each sensor falls off as modeled by the Gaussian distribution $X \sim \mathcal{N}(0, \sigma^2)$. As in U-Air [2] we set σ to be the average distance between any two sensors.
3) *Ordinary Kriging and Random Forest (OK)*: To test against baselines taking into account spatial and temporal features, we implement ordinary Kriging with a linear kernel and a non-streaming random forest method (RF). Ordinary Kriging is a common spatial interpolation method incorporating variation, while random forest is a

TABLE II: Effectiveness of AQI inference.

	Rangiora			Arrowtown			Beijing		
	Accuracy	Kappa	Runtime(s)	Accuracy	Kappa	Runtime(s)	Accuracy	Kappa	Runtime(s)
NC	76.43 (7.10)	45.36 (10.79)	23.66 (0.90)	94.94 (3.48)	**24.81 (7.59)**	30.01 (0.77)	29.99 (3.15)	10.61 (3.22)	9.98 (0.48)
IDW	25.24 (8.94)	3.90 (7.23)	26.96 (1.17)	85.31 (2.96)	14.96 (11.06)	35.81 (0.92)	23.83 (12.68)	6.98 (13.95)	11.64 (0.62)
Gaussian	61.19 (19.35)	28.96 (21.08)	88.92 (4.83)	92.22 (2.53)	17.32 (11.41)	86.10 (3.67)	40.13 (23.13)	27.20 (26.08)	42.57 (2.35)
OK	44.72 (6.90)	15.94 (1.03)	591.36 (58.15)	93.95 (1.75)	10.32 (3.87)	233.32 (13.27)	38.05 (4.80)	23.58 (5.60)	387.09 (23.92)
RF*	76.34 (7.14)	45.02 (10.93)	-	94.94 (3.48)	24.81 (7.58)	-	29.99 (3.15)	10.61 (3.22)	-
ARF	76.38 (7.17)	44.75 (10.83)	410.29 (33.39)	**96.27 (2.63)**	16.57 (8.56)	496.36 (39.66)	36.52 (7.34)	15.39 (9.35)	364.03 (23.02)
AS_b	81.60 (6.57)	59.95 (7.56)	138.80 (9.60)	96.05 (2.83)	13.18 (7.63)	95.31 (5.64)	65.71 (5.77)	57.03 (7.42)	89.51 (5.01)
AS_r	**81.61 (6.62)**	**60.11 (7.65)**	156.39 (12.61)	96.10 (2.75)	12.92 (7.58)	111.17 (9.06)	**65.78 (5.94)**	**57.11 (7.70)**	99.36 (5.81)

*RF is a batch method rather than a streaming method.

tree based method similar to the Hoeffding trees in our system.

4) *Adaptive Random Forest (ARF)*: Finally we compare to a state of the art data stream method, Adaptive Random Forest (ARF) [11]. ARF uses an ensemble of Hoeffding Tree classifiers and can detect and adapt to concept drift, however it does not consider reusing classifiers on multiple stream segments.

To evaluate our proposed repair component, we compare two versions of our AirStream system. A base version not containing the repair component and a full version are denoted as AS_b and AS_r respectively in the following results.

A. *Effectiveness and Robustness of AirStream*

Inference Effectiveness based on Accuracy and Kappa Performance. Table II displays the performance of each method in inferring $PM_{2.5}$ levels. We use the first 20,000 observations for training and report the prequential accuracy for streaming capable systems and the holdout accuracy for the non-streaming RF on the remaining observations. We also report the Kappa statistic, a measure of accuracy above a chance classifier [14]. In *Rangiora* and *Beijing* both versions of AirStream had better performance compared to the baselines. In *Arrowtown* AirStream is competitive with ARF and outperforms all other baselines on prequential accuracy metric. To report performance on unbalanced classes, Figure 8 shows a confusion matrix for *Rangiora* inferences which was averaged across all target sensors, for AirStream and RF. RF was the top performing baseline. RF uses a similar tree based classification method to AirStream. Predictions for both methods are distributed across all $PM_{2.5}$ levels except level 5. Airstream displays better performance at classifying $PM_{2.5}$ levels 0, 1, 3 and 4 and similar performance on level 2 classifications. Neither make any level 5 predictions. The $PM_{2.5}$ distribution in Figure 4 shows there were only 1542 and 71 observations of levels 4 and 5 respectively across all sensors in *Rangiora*. We believe there was not enough training data available to predict these levels accurately.

Synthetic Performance Table III shows the performance of each method on synthetic data sets generated using the RBF and TREE generators. To show the generality of AirStream, these are not interpolation tasks so we do not compare to the IDW and Gaussian interpolation baselines. For each test, 20,000 observations were drawn alternating between two concepts with abrupt and gradual concept drift. The width of the gradual drift was 4000 observations. Results shown are averaged over 100 runs using different concept and sample seeds. To measure classification performance invariant to the length of each concept, we report the Kappa statistic [14] for the 150 observations after each drift. A higher performance in this period indicates better adaption to drift. Note that $F1_c$ here is not a measure of performance, rather it measures condition inference as described in the next section. The synthetic results show AirStream obtains the highest performance. The increased performance seen in AS_r over AS_b indicates that our drift repair algorithm contributes towards this increased performance.

Levels	0	1	2	3	4	5
0	**19652.92**	2041.52	105.916	34.05	1.23	0.00
1	1017.77	**5380.31**	589.55	91.71	1.11	0.00
2	665.08	610.66	**2372.41**	206.30	0.63	0.00
3	50.52	152.60	270.34	**529.34**	2.73	0.00
4	6.90	0.01	3.42	18.15	**3.83**	0.00
5	0.00	0.00	0.00	0.00	0.00	**0.00**

(a) AirStream

Levels	0	1	2	3	4	5
0	**19355.54**	2017.89	307.50	151.31	3.36	0.00
1	2039.86	**4418.30**	392.69	227.34	2.28	0.00
2	383.67	537.63	**2791.94**	141.44	0.41	0.00
3	204.61	296.79	163.90	**337.30**	2.94	0.00
4	0.61	7.15	3.30	19.72	**1.52**	0.00
5	0.00	0.00	0.00	0.00	0.00	**0.00**

(b) Random Forest (RF)

Fig. 8: Average Confusion Matrix of $PM_{2.5}$ Inferences for *Rangiora*

Runtime Analysis Tables **??** and III show the runtime of method compared to the streaming baselines on real and synthetic data. While our proposed repair component incurs a small performance penalty over the base method, overall runtime is much smaller than the state-of-the-art method ARF. As an ensemble method, ARF essentially runs all inactive models at every time step while our repair algorithm runs only K additional inactive model only during the testing period. Our method also uses substantially less memory than ARF, with the repair algorithm contributing only the size of one model to the storage overhead.

Robustness and Sensitivity Analysis. We tested the performance across four parameters, specifically, the window size used for selecting the classifier to use after a drift, the sensitivity of triggering a repair, the number of observations

TABLE III: Effectiveness of inference (Kappa) and condition inference ($F1_c$) on Synthetic data.

	Radial Basis Function				Random Tree			
	Kappa	$F1_c$	Time(s)	Memory(B)	Kappa	$F1_c$	Time(s)	Memory(B)
NC	14.21 (11.98)	48.47 (0)*	4.94 (0.16)	32 (0)	1.14 (4.81)	48.47 (0)*	5.14 (0.15)	32 (0)
OK	-1.75 (3.20)	48.47 (0)*	211.93 (39.17)	7557 (3080)	0.05 (0.76)	48.47 (0)*	134.53 (2.16)	9225 (0.00)
RF	14.01 (12.85)	48.47 (0)*	8.89 (0.27)	2185 (0.00)	1.75 (5.66)	48.47 (0)*	11.49 (0.34)	2185 (0.00)
ARF	59.27 (14.08)	48.28 (6.74)	215.03 (16.96)	2166309 (1270000)	50.68 (8.65)	46.11 (5.34)	282.34 (13.56)	8967269 (3100000)
AS_b	59.44 (11.13)	70.94 (7.28)	33.50 (1.64)	1315897 (410000)	53.27 (8.90)	62.95 (7.86)	44.40 (1.98)	3520354 (705000)
AS_r	59.59 (11.22)	75.89 (8.84)	72.97 (5.71)	1102283 (372000)	54.76 (8.82)	71.00 (7.45)	90.21 (7.25)	3215853 (667000)

* These systems are not adaptive, so $F1_c$ is constant given the same proportion of concepts.

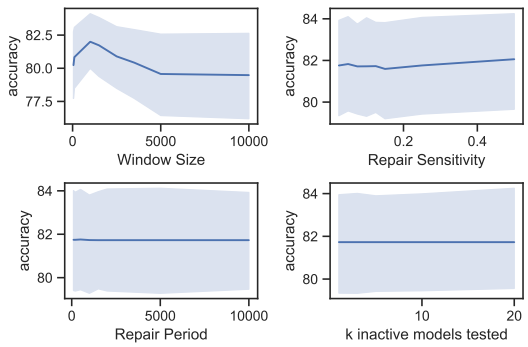

Fig. 9: Sensitivity analysis based on parameter tuning

between each periodic drift repair test and the number of inactive models tested at each drift repair step. Figure 9 shows the sensitivity of our system to these parameters on the *Rangiora* data set, with standard deviation given by 10 repetitions.

B. Effectiveness of Condition Inference

The task of inferring air quality has been shown to be highly reliant on current weather conditions. We hypothesized that a data stream approach would allow changes in these conditions to be adapted to by a system without access to meteorological conditions. In this section we verify whether the changes detected by our system are related to changes in weather.

1) Environmental Data: A different set of environmental conditions was collected for each data set due to availability. We highlight that these features were not available to any system during training or testing, and are only used in post-hoc analysis. Each source of environmental data was discretized into 8 equal density levels.

1) Rangiora: We collected the current wind speed and direction for each observation.
2) Arrowtown: We collected two time based features indicating if the observation was taken during daylight and if the observation was taken during the weekend.
3) Beijing: We used the wind speed and direction, temperature and pressure from the weather station closest to each sensor for each observation.

As discussed in Section V, we calculate the $F1_c$ score between each active model used and each level for every environmental feature to measure the strength of relationship between the two. This is averaged across all levels to find the average $F1_c$ score between AirStream's active models and a given environmental feature.

We also train a random forest classifier to classify the current level of each environmental feature using the current active model. We refer to the accuracy of this system on a given environmental feature as ρ_m in Table IV. We compare this to the accuracy of a random forest classifier trained to predict the current level using current $PM_{2.5}$ readings, referred to as ρ_f, to find the additional performance in condition inference gained by using AirStream. To maximise the performance of this baseline we use all sensors with no masking so ρ_f displays no variation across repetitions.

Table IV shows ρ_f, ρ_m and $F1_c$ for each environmental condition for the data sets they are available. The classifier trained on the active model of AirStream has a higher predictive performance than the classifier trained on recent sensor readings in inferring current wind direction and speed in *Rangiora*. We can also see $F1_c$ scores of approximately 0.47 for both conditions. This indicates some level of predictive ability from the models produced by our system. This adds validity to the hypothesis that our system can react to changes in environmental conditions without requiring additional input features. However, the results from the *Beijing* data set are less ideal. We note that the AirStream active model does not correspond particularly highly with temperature, pressure, wind direction and wind speed in *Beijing*. We hypothesize that this is due to a mismatch between the speed of change in these conditions and observation frequency, which was 1 hour in *Beijing* compared to 1 minute in other data sets.

Table III shows similar condition inference results on synthetic data streams. The $F1_c$ score here compares the state of the system to the known generating function each observation is drawn from. For NC, OK and RF the system state is constant. We see the expected result that a constant state does not correspond strongly to a dynamic stream with an $F1_c$ score of only 0.48. For ARF we take the state of the system to be the ensemble model with the highest contribution to the vote. We see that this has a weaker relationship than the constant state, indicating the adaptions made by ARF cannot be analysed to infer conditions. AirStream presents substantially higher $F1_c$ scores, indicating a stronger relationship between state and stream conditions. The higher result for AS_r indicates our repair algorithm provides a more robust stream model.

2) Visualisation: Changes in environmental conditions come from complex interactions between many factors, so

TABLE IV: Condition inference on Real Data sets.

Data Set	Condition	P_f	P_m	$F1_c$
Rangiora	Wind Direction	34.87*	50.39 (7.93)	0.47 (0.06)
Rangiora	Wind Speed	33.26*	49.41 (7.97)	0.47 (0.06)
Arrowtown	IsDaytime	74.01*	74.46 (3.72)	0.35 (0.09)
Arrowtown	IsWeekend	71.12*	67.81 (6.51)	0.39 (0.08)
Beijing	Wind Direction	14.52*	14.70 (0.40)	0.21 (0.02)
Beijing	Wind Speed	35.92*	22.29 (2.89)	0.31 (0.04)
Beijing	Temperature	34.03*	27.02 (4.14)	0.33 (0.03)
Beijing	Pressure	32.68*	26.30 (3.97)	0.33 (0.03)

* ρ_f is trained on all data points to give an optimal baseline so displays no variation across repetitions

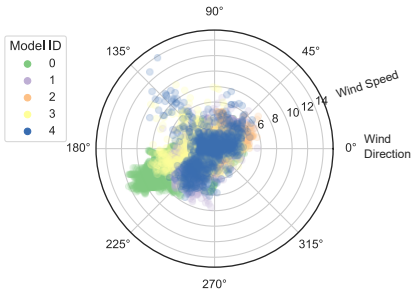

Fig. 10: Weather conditions compared to system model. Angle of each observation is given by wind direction, magnitude from (0, 0) is given by wind speed and color is given by active model (referred to by a unique ID).

it is unlikely the models constructed by our system will perfectly match with any one feature. For example, rather than a single model relating to northerly wind, our system may find a model relating to northerly wind at high wind speed and another relating to northerly wind at low wind speeds. The previous evaluation does not consider the relationship between the model used and *sets* of environmental conditions. Figure 10 visualises the relationship between active model and the combination of wind speed and direction. We plot each observation in the *Rangiora* data set as the vector with direction given by wind direction and magnitude given by wind speed. We color each data point by the active model chosen by the system (referred to by an unique ID). Certain combinations of wind speed and direction can be seen to be related to certain active models. For example strong south westerly wind is largely classified by Model ID 0 in Figure 10.

VII. Related Work

Recent work on air quality prediction has focused on urban locations, incorporating the many sources of environmental features available in these areas to improve performance. We instead investigate rural locations where these features are not available.

Zheng et al. [2], [3] separated spatial, temporal and meteorological features, feeding subsets of features into a spatial ANN model, a temporal linear regression or conditional random field model, and an inflection prediction model. The outputs of these models are merged based on current meteorological features. Meteorological features used include category of weather (sunny, overcast, *etc.*), humidity, wind speed and wind direction, as well as the forecast weather at a particular location. The authors noted that the use of these conditions is especially important in detecting inflection points where prediction patterns change. The link between inflection points and changes in weather, is however, not investigated. Yi et al. [15] also utilized meteorological features, integrating them with spatial and temporal features using a deep fusion network. They noted that these 'indirect' features affect spatial and temporal transmission patterns, however, they do not attempt to detect such changes explicitly. Cheng et al. [6] incorporated meteorological features, weather, temperature, pressure, humidity and wind as well as points of interest and road network features into an attention based neural network model. They investigated feature importance, but did not compare feature importance under differing weather conditions. The attention mechanism is similar to our drift adaption method, whereby we attempt to find the most relevant features for each point in time. A critical difference is our method performs this in a low-information environment. Hsieh et al. [16] investigated an offline method of constructing an 'AQI Affinity graph' measuring the relationship between sensor readings over time. They used additional traffic and point of interest features to find similar sensors when constructing this graph. Change in spatial or temporal relationships is also not considered. Shang et al. [17] investigated inferring traffic pollution from road network features, while Devarakonda et al. [18] monitored traffic pollution using mobile sensors. These approaches are not applicable in the rural environments studied here. Low traffic volume in these environments means wood burning is a much more important source of air pollution than vehicles.

Similar to Zheng et al. [3], we attempt to detect and adapt to inflection points, however we do this without knowledge of the relevant weather conditions. This allows our system to be used in more locations and without relying on hand picked weather features. This also allows us to adapt to drift in features not previously investigated, such as changes in sensor sensitivity.

Other methods attempted condition inference with only spatial and/or temporal readings. Hu et al. [19] developed a 3D $PM_{2.5}$ interpolation system. A random walk approach models pollution transmission between 3D grid cells, with transmission rates learned from data. No change detection was implemented, and their results showed large performance variation as weather changed. Hu et al. [20] used a Gaussian interpolation to infer $PM_{2.5}$ readings without the use of environmental features. Similar to previous research, no method of detecting or adapting to change were investigated. Li et al. [21] investigated a method of mining causality patterns between sensors to determine propagation patterns and locate sources. The timing of uptrend events is matched across sensors to determine causality. In this research, no considerations were made to capture changes in these causality patterns over time. They concluded that the propagation patterns they found had no relationship to meteorological conditions.

VIII. Conclusion

We proposed AirStream, a data stream based classification system able to detect and adapt to changes in unknown environmental conditions. By applying AirStream to the air quality inference task, we show that concept drift detection can be utilized to create systems capable of adaptation without large data requirements. We develop a repair algorithm to increase the robustness of this adaption, allowing a mapping between the state of the system and environmental conditions. Our evaluation shows that AirStream produces high performance $PM_{2.5}$ inference in two rural towns which lack meteorological monitoring, outperforming six baselines. Analysis of meteorological conditions in one of the locations shows that the adaptions made by AirStream can be used to predict wind speed and direction with above 48% accuracy, 14.8% higher than using air quality readings alone. We believe the active model used by AirStream has the potential to allow the inference of environmental conditions in locations where they were previously unavailable. For future work we plan to leverage the changes detected by AirStream to further improve inference performance, for example, by automatically tuning system parameters based on learned transition patterns. We also plan to investigate the generality of AirStream by applying it to other tasks affected by change in unknown features.

References

[1] J. Lelieveld, K. Klingmüller, A. Pozzer, U. Pöschl, M. Fnais, A. Daiber, and T. Münzel, "Cardiovascular disease burden from ambient air pollution in Europe reassessed using novel hazard ratio functions," *European Heart Journal*, vol. 40, no. 20, pp. 1590–1596, May 2019.

[2] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*. ACM, 2013, pp. 1436–1444.

[3] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, "Forecasting fine-grained air quality based on big data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 2267–2276.

[4] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[5] D. W. Wong, L. Yuan, and S. A. Perlin, "Comparison of spatial interpolation methods for the estimation of air quality data," *Journal of Exposure Science & Environmental Epidemiology*, vol. 14, no. 5, pp. 404–415, 2004.

[6] W. Cheng, Y. Shen, Y. Zhu, and L. Huang, "A neural attention model for urban air quality inference: Learning the weights of monitoring stations," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[7] H. Huggard, Y. S. Koh, P. Riddle, and G. Olivares, "Predicting air quality from low-cost sensor measurements," in *Australasian Conference on Data Mining*. Springer, 2018, pp. 94–106.

[8] United States Environmental Protection Agency, "Revised air quality standards for particle pollution and updates to the air quality index (aqi)," United States Environmental Protection Agency, Tech. Rep., 2012.

[9] S. Zhang, B. Guo, A. Dong, J. He, Z. Xu, and S. X. Chen, "Cautionary tales on air-quality improvement in beijing," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2205, p. 20170457, 2017.

[10] J. Gama, R. Rocha, and P. Medas, "Accurate decision trees for mining high-speed data streams," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 523–528.

[11] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, no. 9-10, pp. 1469–1495, 2017.

[12] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 2007, pp. 443–448.

[13] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: massive online analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 1601–1604, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?id=1859903

[14] I. Žliobaitė, A. Bifet, J. Read, B. Pfahringer, and G. Holmes, "Evaluation methods and decision theory for classification of streaming data with temporal dependence," *Machine Learning*, vol. 98, no. 3, pp. 455–482, 2015.

[15] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng, "Deep distributed fusion network for air quality prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2018, pp. 965–973.

[16] H.-P. Hsieh, S.-D. Lin, and Y. Zheng, "Inferring air quality for station location recommendation based on urban big data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 437–446.

[17] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu, "Inferring gas consumption and pollution emission of vehicles throughout a city," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 1027–1036.

[18] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, "Real-time air quality monitoring through mobile sensing in metropolitan areas," in *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*. ACM, 2013, p. 15.

[19] Y. Hu, G. Dai, J. Fan, Y. Wu, and H. Zhang, "Blueaer: A fine-grained urban pm2. 5 3d monitoring system using mobile sensing," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[20] Z. Hu, Z. Bai, K. Bian, T. Wang, and L. Song, "Implementation and optimization of real-time fine-grained air quality sensing networks in smart city," in *IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[21] X. Li, Y. Cheng, G. Cong, and L. Chen, "Discovering pollution sources and propagation patterns in urban area," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1863–1872.