Automatic Quality Control of Crowdsourced Rainfall Data with Multiple Noises: A Machine Learning Approach

Geng Niu [a,b,c], Pan Yang [c*], Yi Zheng [b,d*], Ximing Cai [c], Huapeng Qin [a]

[a] Key laboratory for Urban Habitat Environmental Science and Technology, School of Environmental and Energy, Peking University Shenzhen Graduate School, Shenzhen 518055, Guangdong Province, China

[b] School of Environmental Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, Guangdong Province, China

[c] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Champaign, Illinois, USA

[d] Shenzhen Municipal Engineering Lab of Environmental IoT Technologies, Southern University of Science and Technology, Shenzhen 518055, Guangdong Province, China

*Corresponding authors:
Pan Yang, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Champaign, Illinois, USA. (pyangac@illinois.edu)
Yi Zheng, School of Environmental Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, Guangdong Province, China. (zhengy@sustech.edu.cn)

Abstract

In geophysics, crowdsourcing is an emerging non-traditional environmental monitoring approach that encourages contributions of data from individual citizens. Because of their reliance on undertrained citizens and imprecise low-cost sensors, crowdsourced data applications suffer from different types of noises that can deteriorate the overall monitoring accuracy. In this study, we propose a machine learning approach for automatic Crowdsourced data Quality Control (CSQC) by detecting and removing noisy data points in spatially and temporally discrete crowdsourced observations. We design a set of features from the original and interpolated rainfall data, and apply them to train and test the CSQC models based on both supervised and non-supervised machine learning algorithms. Performances of the CSQC models under various scenarios assuming no further retraining are also tested (hereafter referred to as transferability). The results based on synthetic but realistic data show that the CSQC model can significantly reduce the overall rainfall estimation error. Under the stationary assumption, CSQC models based on both supervised and unsupervised algorithms can have decent performances in noisy data identification and overall rainfall estimation error reduction; however, if the model is transferred to other cities with different rainfall structure or noise composition (without retraining), the supervised Multi-Layer Perceptrons (MLPs) turns out to be the best performing one.

Key points:

- A machine learning-based quality control approach is proposed for crowdsourced rainfall data with discontinuity in both time and space
- Quality control models are based on both supervised and unsupervised learning algorithms
- Performances of quality control models under various scenarios with and without retraining are tested
- The supervised multi-layer perceptron turns out to be the best performing algorithm under almost all scenarios

1. Introduction

In the recent years, crowdsourcing, an alternative data acquisition approach that involves the collection of data from individual citizens through the internet, social media, and smartphones, has been increasingly investigated, especially in the field of geophysics (Ebert et al., 2018; Wu & Wang, 2019). In comparison to the costly traditional geophysical data collection approach (that largely relies on expensive professional instruments) adopted by researchers and governments (de Vos et al., 2019), the crowdsourced approach uses human judgments or low-cost sensors of common citizens as the data source. It thus offers a way of obtaining massive data cost-effectively. In some developing countries, crowdsourcing can be even a major source of geophysical data (Pingali, 2017).

The crowdsourcing approach has been demonstrated to increase the spatial and temporal representativeness of geophysical observation network, and has been applied in a broad range of areas, e.g., climate research (Meier et al., 2017), air quality (Schneider et al., 2017), ecology (Hunt et al., 2017), geography (Fan et al., 2016), and especially, rainfall. In the past two decades, the number of personal weather stations (PWS) in the US has been growing exponentially from nearly 2,000 in 2001 to almost 100,000 in 2019 (Chen et al., 2019), significantly outnumbering the 9,300 professional rain gauges operated or managed by National Oceanic and Atmospheric Administration (NOAA) (Durre et al., 2013). In recent years, crowdsourcing-based rainfall monitoring is becoming even more attractive (Haklay, 2013) because of the continuous developments in information extraction from smartphones (Guo et al., 2019), low-cost sensors (e.g., surveillance cameras) (Jiang et al., 2019), microwave links (Overeem et al., 2016), and moving cars (Rabiei et al., 2016). The utilization of crowdsourced precipitation data has provided an essential supplement to traditional measurements based on ground gauges and radars (Fencl et al., 2017; Gosset et al., 2016).

However, there can be significant uncertainties surrounding the quality of crowdsourced data, and proper quality control is required to filter out crowdsourced observations with overly large errors (hereafter referred as the noisy data) (Foody et al., 2013; Steger et al., 2017; Walker et al., 2016). Meanwhile, crowdsourced rainfall data are heterogeneous and unstructured in nature, and therefore require specialized methods to handle the noisy data, improve data quality, and produce useful information for different applications (Zheng et al., 2018). In rainfall monitoring, instrumental errors, compromised setup, data processing issues, operation noise from untrained crowdsourced participants, and sampling error can all lead to noise observations (de Vos et al., 2019; Walker et al., 2016). These can be attributed to anthropogenic factors (e.g., incorrect location report) and equipment errors (e.g., camera lens failure). For example, in many cases, twitter data are considered to be lack of credibility as only 1%-2% of the data are geo-labeled and readily interpretable (Middleton et al., 2013; Palen & Anderson, 2016). Similarly, the data of PWS are more error-prone than traditional rain gauges as they are usually subject to installation and maintenance deficits (e.g., devices may clog after windy weather) (Bell et al., 2015).

In geophysical studies, many approaches have been proposed to improve the accuracy and quality of the crowdsourced data. Some compare the crowdsourced data with expert judgments or a gold-standard data set (Kazai et al., 2013; Zheng et al., 2018). But such a method is considered to be not scalable for two reasons: i) the limited number of experts available when compared to a large number of crowdsourced participants, and ii) the benchmarking database might be outdated (Goodchild & Li, 2012). Others identify the noisy data from crowdsourced observations by a set of preset rules. For example, de Vos et al. (2019) proposed a method to detect and filter four types of noises from PWS observations through a set of if-then rules. These rules are based on a simple validity test and comparison with adjacent observations, and specific thresholds of these rules are calibrated based on a large set of historical data. The method is easy to implement but too simple to be applicable for complex crowdsourced cases with multiple sources (rather than only PWS) of observations and unknown uncertainties.

Other more advanced studies adopt machine learning (ML) approaches to identify noisy crowdsourced observations (Aggarwal, 2015; Goldstein & Uchida, 2016). The advantage of the machine learning approach lies in its ability to reliably approximate the complex, nonlinear relationship between the quality of a data point and its associated features. It also has the advantages of flexibility and scalability to adapt to different application scenarios, as well as the ability to avoid overly subjective judgment on the thresholds of quality control rules (Allahbakhsh et al., 2013; Alpaydin, 2014; Lease, 2011; Leigh et al., 2019). For instance, Moatar et al. (1999) applied artificial neural networks (ANNs, a supervised learning model) to quality control a river water PH estimate model. The ANNs model was used for detecting abnormal values, discontinuities, and drifts in PH measurement screening. Talagala et al. (2019) introduced an unsupervised learning approach aimed to detect anomalies (including sudden spikes, isolated drops, and level shifts) in in-situ water quality (turbidity, conductivity, and river level) monitoring data. Their study emphasized the advantages of unsupervised learning as it does not require labeled data for training and can be readily transferable to other similar scenarios without additional retrain, while such an ability (transferability) is untested.

The previous studies have developed sound, initial steps for effective quality control of crowdsourced data. However, they are only applicable to fixed-point sensors with continuous observation and might not be effective in the quality control of more general crowdsourced rainfall observations from both mobile and fix-point sensors that collect data at various frequencies (Yang & Ng, 2017). The design of a quality control algorithm for the general crowdsourced rainfall data is a non-trivial task. The crowdsourced data could be non-continuous in both space and time, and it is difficult to extract directly useful information from adjacent or historical observations as the previous studies do. Moreover, there is a need to systematically compare the performances of supervised and unsupervised learning techniques in identifying noisy crowdsourced observations, especially in terms of their applicability to other locations and/or scenarios without further retraining (i.e., the transferability). In general, among the two approaches, supervised learning should be performing better with conditions similar to its training data, but its performance might be compromised when the input dataset is not seen by the model during the training phase; unsupervised learning usually could have consistent performances with different sets of input data, but it generally does not perform better than the supervised learning algorithms when there exist high-quality training labels (Mohammady et al., 2015; Sathya & Abraham, 2013). It is unclear which approach is performing better in terms of transferability and, therefore, should be selected as the recommended practice.

In this study, we develop a machine learning-based quality control mechanism to detect noisy data in general crowdsourced rainfall observations that consist of non-continuous data in both space and time. A set of supervised and unsupervised algorithms are trained and tested for their ability to identify the noisy points, as well as their performances with unseen inputs/scenarios without retraining. The testing is made with synthetic but realistic data assuming climate conditions from three major U.S. cities with different rainfall patterns. While the machine learning approach has been applied in identifying noise in environmental observations, according to our knowledge, what we propose is the first to propose an automatic algorithm to detect noise in a general type of crowdsourced observations. This study is also the first in testing the transferability of different supervised and non-supervised learning algorithms for data quality control. Given the increasing number of adoptions of crowdsourced-based environmental monitoring, this study provides a timely contribution to this specific area by introducing a robust and readily transferable algorithm to identify the largely unavoidable noisy points in data from those crowdsourcing projects.

2. Methodology

In this study, we propose a machine learning approach for Crowdsourced data Quality Control (CSQC), i.e., for detecting and removing noisy points in a general rainfall crowdsourced model that consists of discontinuous data in both space and time. Our procedure can be generalized into **four** main steps shown in Figure 1:

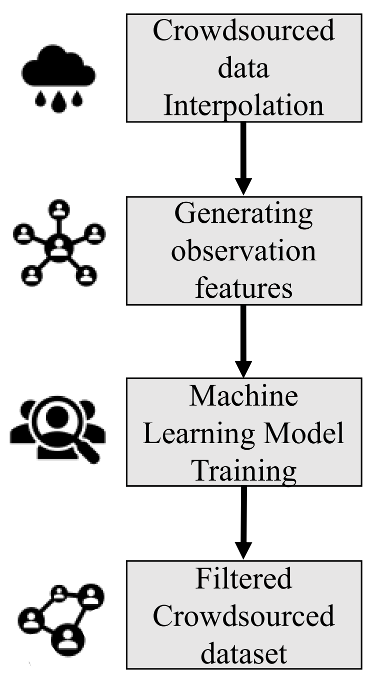

Figure 1. General framework of CSQC procedure for noisy point identification in crowdsourced rainfall observation.

We first interpolate the crowdsourced rainfall data into a crowdsourcing rainfall field with gridded estimation of rainfall intensities. We then generate a set of input features from the crowdsourcing rainfall field and the original crowdsourced data, based on which the machine learning CSQC models are trained and tested. The CSQC model produces two labels for the crowdsourced observations: regular and noisy. As a final step, we filter the noisy observations and compare the original and quality-controlled observations via various performance metrics.

2.1. Feature extraction

We assume the crowdsourced rainfall data are coming from both fixed-point sensors and mobile citizens. To compensate the discontinuous information in crowdsourced data, we first interpolate (through nearest-neighbor interpolation, Jones et al., 2001) rainfall observations collected within a short time-duration into a spatial rainfall field (hereafter referred as the crowdsourcing rainfall field), assuming rainfall intensities during this short period of time would not change much. Features for the CSQC algorithm are then extracted from both the original and interpolated crowdsourced data. Two sets of features, i.e., the windows-based and distance-based features, are generated for each crowdsourced observation.

With the crowdsourcing rainfall field, we construct a set of statistics from windows with various sizes centered around the grid cells where crowdsourced observations locate in (i.e., black squares in Figure 2), hereafter referred as window-based features. The constructed statistics include window maximum ($I_{\mathrm{max,W}}$), minimum ($I_{\mathrm{min,W}}$), range (window maximum minus minimum, ($I_{\mathrm{ran,W}}$)), average ($I_{\mathrm{mean,W}}$), standard deviation ($I_{\mathrm{std,W}}$), variance ($I_{\mathrm{var,W}}$), absolute deviation (i.e., the difference between crowdsourced observation and window average, AD_{W}), and relative deviation (i.e., the ratio of absolute deviation to window average, RD_{W}) of the crowdsourcing rainfall field. We also include the lag-1 correlations between the estimated rainfall intensities (within the window) of the current time step and its previous time step as additional window-based features. A total of five different window lengths are selected ranging from 3 to 11 grid cells with an interval of 2 grid cells. It should be noted that, the window is truncated based on the study region boundary if it exceeds the study region (i.e., the crowdsourced observation locates in the edge of the study region).

We further generate a set of distance-based features from the original crowdsourced observations. More specifically, we draw a circle with a specific radius (6 grid cells) around the target crowdsourced observation (the black circle in Figure 2), and calculate a set a statistics from the crowdsourced observations locate within the circle. The statistics include circle absolute deviation (i.e., difference between the target crowdsourced observation and the circle average, AD_{C}), range (i.e., circle maximum minus minimum, $I_{\mathrm{ran,C}}$), and the absolute value of difference between range and sample point value($|I_{sample.C} - I_{\mathrm{ran,C}}|$) which indicates whether the sample point value is close to the extreme value of the interval.

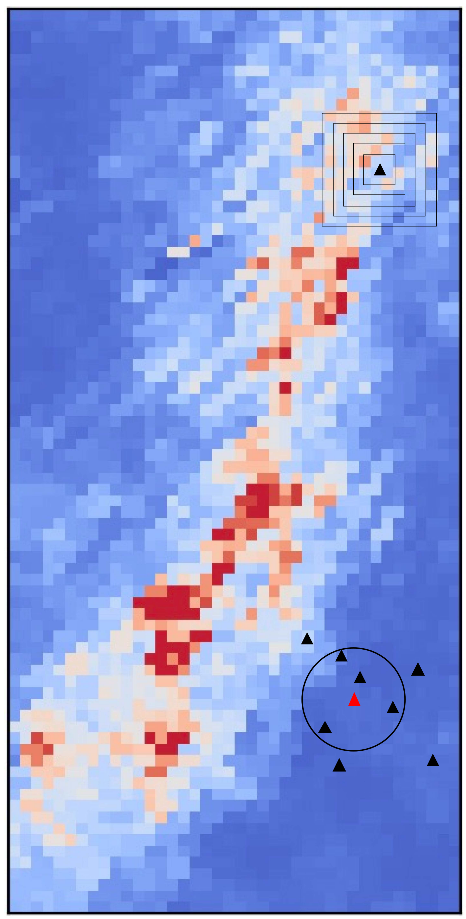

Figure 2. A graphical schematic of window-based and distance-based features extraction, a set of squares with a same center denotes the window-based features selection, the circle represents the distance-based features selection.

2.2. Noisy data identification based on machine learning models

In this study, we develop and test two supervised (*k*NN: *k*-Nearest Neighbors; MLPs: Multi-layer Perception) and two unsupervised (iForest: Isolation Forest; K-means clustering) machine learning algorithms regarding their ability in identifying noisy data from a general type of crowdsourced observations. Inputs to the ML models are the features extracted in section 2.1, and the target is the binary label of the crowdsourced observation: noisy and regular. The supervised learning algorithm assumes a set of pre-labeled crowdsourced data, based on which a classification rule is trained. The unsupervised learning algorithm assumed no such pre-labeled dataset and learns the division of noisy and regular observations from only a dataset of input features.

k-Nearest Neighbor algorithm (*k*NN) is an instance-based model that classifies a target instance based on its *k* nearest pre-labeled instances in the high dimensional space defined by input features (Zhang et al., 2017). *k*NN is an easy to implement albeit highly efficient supervised learning algorithm that has been widely applied in a wide range of studies (Bhatia, 2010; Peterson, 2009; Saini et al., 2013) It is also a flexible model that makes no assumption about the form of input-output relationships or distributions. Multi-layer perceptrons (MLPs) is a type of artificial neural network (ANN) widely applied in many fields too (Altunkaynak & Strom, 2009; Ding et al., 2013; Sahoo et al., 2017). MLPs consists of stacked layers (one input layer, one or more hidden layers, and one output layer) of interconnected nodes. The nodes (i.e., Neurons) are basic components of MLPs that treat outputs from previous layer's neurons through an activation function. The flexibility in the choice of network architecture (i.e., number of layers and number of neurons in each layer) and activation function gives MLPs a learning ability to approximate complex nonlinear relationships in high precision.

Isolation Forest (iForest) is a tree-based unsupervised learning algorithm (Liu et al., 2008). It is developed based on the idea that outliers should be scarce and abnormal, and thus, when compared to non-outliers, they are easier to be isolated through a set of random partitioning trees. iForest is an algorithm specially designed for detecting outlier/noise, and it also holds the advantage of easy implementation and high computational efficiency. K-Means is an unsupervised clustering algorithm, where each cluster is defined based on the cluster center located in the input space. Given a training sample and a pre-specified number of clusters, K-Means automatically finds the cluster centers through an iterative approach, and then assigns instances to their closest clusters. K-Means is well-suited for large sample clustering and has widely used in different fields (Kanungo et al., 2002) including noise detection (Lima et al., 2010).

To train the supervised learning and unsupervised learning models, a random training-testing splitting is used with 70% of the collected data for training and 30% for testing. The training is performed with a five-fold cross-validation to avoid overfitting. Before training, we implement a min-max scaling to normalize input features into consistent ranges. Given the large number of features extracted in section 2.1, we adopt a feature selection process to identify a parsimonious model that is relatively resistant to over-fitting. In the process, the features with the highest importance measured by the Extra Tree algorithm (Geurts et al., 2006) are selected. The number of most important features is identified through a trial and error method as the one with the highest binary classification accuracy. Based on preliminary analysis, 5 features are selected for the unsupervised learning algorithms (Table S2 in the SI) and all features extracted in section 2.1 are selected for the two supervised learning algorithms.

Each of the supervised and unsupervised learning algorithms is associated with several hyper-parameters that need to be specified. A list of the hyper-parameters for each algorithm and their specific meanings are shown in Table S1 in the SI. In this study, we use an exhaustive cross-validated grid-search to identify the optimal combination of hyper-parameter values over pre-specified ranges. The accuracy score evaluated with the validation set is selected as the performance measure of the grid-search algorithm (see Table S1 in SI for the optimal hyper-parameter values selected in this study). Further structure details and relevant settings (including activation function, optimization method, etc.) of MLPs can be referred to section I of the SI. The supervised and unsupervised algorithms and the grid-search hyper-parameter optimization method are implemented with the ‘scikit-learn’ package (Pedregosa et al., 2011) in Python.

2.3. Case studies

2.3.1 Study area and data

In this study, three cities with significantly different climatic conditions are included: San Diego, Chicago, and Miami. San Diego has a Mediterranean climate with annual average rainfall ranging 230-330 mm; Chicago shows a typical hot-summer humid continental climate with most of its rainfall brought by severe and short thunderstorms, and its average annual rainfall reaches 965 mm; finally, Miami has a tropical monsoon climate, and most of its 1,572 mm annual rainfall comes during June - October. The radar data collected from the Next Generation Weather Radar (NEXRAD) system (NOAA, 2013; available at https://www.ncdc.noaa.gov/data-access/radar-data/nexrad) are used as the 'ground-truth' rainfall data in the three cities. The radar data has a 500 m × 500 m × 5 mins resolution and covers a 40 × 20 km^2 space. Data from San Diego is used to train the CSQC model with supervised and unsupervised algorithms. Further, to verify the robustness of the CSQC procedure, the trained model is directly applied and tested with rainfall data from Chicago and Miami. Table 1 shows detailed statistics of the selected storm events from the three cities.

Table 1. Summary statistics of rainfall events, the statistics are calculated from observed radar data from the Next Generation Weather Radar (NEXRAD) system.

	Date of event	Timing of corresponding event	Average rainfall intensity (mm/hr)	Standard deviation of rainfall intensity (mm/hr)
San Diego	2014/12/12	23:00-24:00 UTC	4.23	7.68
	2015/05/08	21:30-22:30 UTC	3.12	5.91
	2015/09/15	21:35-22:35 UTC	1.23	2.37
	2015/10/05	21:00-22:00 UTC	4.96	11.89
	2015/11/04	05:05-06:05 UTC	1.46	4.69
City of the Chicago	2013/04/18	06:30-07:30 UTC	26.36	34.02
	2013/05/20	05:00-06:30 UTC	8.74	13.10
	2013/05/29	03:50-05:00 UTC	4.17	28.71
City of the Miami	2013/04/30	21:10-22:20 UTC	12.38	36.23
	2013/05/01	21:30-22:50 UTC	7.42	31.82
	2013/05/20	10:00-23:50 UTC	16.88	35.92

2.3.2 Synthetical data generation

We test the CSQC model developed in sections 2.1 and 2.2 through a set of synthetic but realistic scenarios. In those scenarios, a set of 'ground-truth' rainfall fields are assumed as the radar rainfall data collected in section 2.3.1. Given that, we assume the crowdsourced observations are taken by the participants at random locations and time points. The synthetic crowdsourced data are generated by adding an observation error to the 'ground-truth' rainfall intensity at locations and time where crowdsourced observations are taking place, following Yang & Ng (2017):

$$E \sim N\left(\beta_e \cdot I_{true},\ (\alpha_e \cdot I_{true})^2\right) \quad (1)$$

where *E* denotes the observation error, *N* () a normal distribution, α_e the coefficient of variation, and β_e the coefficient of bias. We further assume that crowdsourced observations are provided by two types of participants, i.e., regular participants and low-performing participants. For observations from regular participants, we adopt a suggestion from Mazzoleni et al. (2017) and set α_e as a random variable following uniform distribution ranging from 0.1 to 0.2, and β_e a uniformly distributed random variable ranging from -0.15 to 0.15. α_e and β_e values for observations from low-performing participants are having larger values, and we test ten scenarios of their values as shown in Table 1.

Distributions of real-world observation errors are more complex and might be skewed (Dennis et al., 2006). To test the impact of a skewed observation error distribution on the performances of CSQC algorithms, we generate a separate set of synthetic crowdsourced observations with a random error following the Wald distribution. Wald distribution is a special case of inverse Gaussian distribution with its shape flexibly controlled by two parameters: the mean μ and the scale λ. An illustration for the probability density functions of Wald distribution under different μ and λ values are shown in Figure S8 in SI. In this study, we fix the ratio of μ and λ to be 2 to make the Wald distribution positively skewed. To make a fair comparison between the Normal distribution and Wald distribution scenarios, we manipulate the value of μ and λ to generate a set of Wald distributions with their means and standard deviations equal to the values shown in Table 1.

We then manually label the generated crowdsourced observations into 'regular observation' and 'noisy observation' for the supervised learning algorithms. An observation is labeled noisy using two criteria: a relative error criterion and an absolute error criterion. The former follows Bauer et al. (2002), which identifies an observation as noisy only if its value is smaller than 50% or larger than 150% of the 'ground truth'; the latter requires the noisy observation to have an error at least larger than 0.1 (mm/hr). The rule to identify noisy observation can be formulated as:

$$|I_{obs} - I_{true}| > 0.5 \times I_{true} \;\; \& \;\; |I_{obs} - I_{true}| > 0.1 \text{ (mm/hr)} \qquad (2)$$

where I_{obs} is the observed rainfall intensity, and I_{true} the 'ground truth' intensity. If a crowdsourced observation follows the rule specified in equation (5), it will be labeled as noisy; otherwise it will be labeled as regular.

2.3.3. Scenario design

Table 2. Noise related and Crowdsourced density scenarios setting

Noise Level scenarios

	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
β_o	[-0.35, 0.45]	[-0.35, 0.55]	[-0.35, 0.65]	[-0.35, 0.75]	[-0.35, 0.85]	[-0.35, 0.95]	[-0.35, 1.05]	[-0.35, 1.15]	[-0.35, 1.25]	[-0.35, 1.35]
α_o	[0.7, 1.3]	[0.8, 1.4]	[0.9, 1.5]	[1.0, 1.6]	[1.1, 1.7]	[1.2, 1.8]	[1.3, 1.9]	[1.4, 2.0]	[1.5, 2.1]	[1.6, 2.2]
Noise amount	35%									
CS density (km^2*hr)	0.75									

Noise Amount scenarios

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
β_o	[-0.35, 0.95]									
α_o	[1.2, 1.8]									
Noise amount	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
CS density (km^2*hr)	0.75									

CS Density scenarios

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
β_o	[-0.35, 0.95]									
α_o	[1.2, 1.8]									
Noise amount	35%									
CS density (km^2*hr)	0.375	0.750	1.125	1.500	1.875	2.250	2.625	3	3.375	3.750

For each set of 'ground truth' rainfall data, we generate a series of scenarios which are described by three variables: noise level, noise amount, and crowdsourcing density. The noise level focuses on the magnitude of noise defined by the α_e and β_e values (equation 1) of low-performing participants. The noise amount equals to the portion of low-performing participants in the whole crowdsourcing dataset, and the crowdsourcing density is the total number of crowdsourced observations per time step in the study area. The three factors could potentially alter the distribution of crowdsourced observation errors and therefore have an impact on the performances of different CSQC algorithms. Among all the scenarios, we set a benchmark scenario with $\alpha_e \in$ [1.2, 1.8], $\beta_e \in$ [-0.35, 0.95], noise amount equal to 35%, and crowdsourcing density equal to 0.75/(km^2*hr). In addition, we generate a series of noise level scenarios, noise amount scenarios, and crowdsourcing density scenarios by varying one parameter of the benchmark scenario at a time (Table 2).

2.4. Sensitivity analysis and model transfer

In this study, we test different CSQC algorithms under two different settings: i) a sensitivity analysis that retrains the CSQC algorithms under different scenarios, and ii) a model transferability test that directly apply a trained CSQC model to different scenarios without re-train. Under the sensitivity analysis setting, the four CSQC algorithms in section 2.2 are trained and tested for each of the crowdsourcing scenarios in Table 1 by assuming the 'ground truth' rainfall field from San Diego. Under the model transfer setting, the CSQC algorithms are first trained with the benchmark scenario in San Diego and subsequently tested without re-train under all crowdsourcing scenarios in Table 2 in San Diego, as well as in Chicago and Miami where climate conditions are significantly different from San Diego.

Results from the sensitivity analysis could be helpful for understanding the impact of Noise Level, Noise Amount, and CS Density on performances of different CSQC algorithms, and results from the model transferability test provide information for the generalization properties of the CSQC models. Combing the two information, practical guidance for the choice of CSQC algorithm with high performance and flexibility could be generated for practitioners.

2.5. Comparison statistics

Two types of statistics are used in this study: statistics for noise identification and for rainfall field estimation. Identification of noisy crowdsourced observation is a typical binary classification or clustering task, and we use four statistics to quantify the binary classification performances of different CSQC algorithms, namely the classification accuracy, negative predictive value (*NPV*), positive predictive value (*PPV*), and area under the receiver operating characteristic curve (*AUC*).

Accuracy measurement explains the overall effectiveness of a classifier in making correct predictions, and is calculated as:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (3)$$

where *FP* denotes the number of false positive instances; *FN* the number of false negative instances; *TP* the number of true positive instances; and *TN* the number of true negative instances. Here a positive instance represents a noisy label for the crowdsourced observation, and negative instance the regular label.

Negative Predictive Value (*NPV*) and Positive Predictive Value (*PPV*) (Ranawana & Palade 2006) provide more detailed information about the performance of a model in predicting positive and negative instances:

$$NPV = \frac{TN}{FN+TN} \quad (4)$$

$$PPV = \frac{TP}{TP+FP} \quad (5)$$

Area Under Curve (*AUC*) is a robust classification performance metric (Fawcett, 2006) that measures the area under the *ROC* (Receiver Operating Characteristic) curve, which plots the True Positive Rate (*TPR*) against False Positive Rate (*FPR*) under various discrimination threshold settings.

$$TPR = \frac{TP}{TP+FN} \quad (6)$$

$$FPR = \frac{FP}{FP+TN} \quad (7)$$

With an range from 0 to 1, an AUC value equals to 0.5 represents a classifier equivalent to random guess, and an *AUC* value equals to 1 represents a perfect classifier.

One of the purposes of this study is to examine the effectiveness of the trained machine learning model on improving crowdsourced data's ability in representing the 'ground-truth' rainfall field. Therefore, we use two rainfall field related statistics to measure the performances of different

CSQC algorithms: i) root mean square error of rainfall field estimated from crowdsourced observations (*RMSE*), and ii) relative error in the areal average rainfall estimated from crowdsourced observations (*REAA*).

RMSE represents the ability of the estimated rainfall field to capture the storm's rainfall variability on a small spatiotemporal scale and is defined as:

$$RMSE = \sqrt{\frac{1}{G_X G_Y G_{TR}} \sum_{x=1}^{G_X} \sum_{y=1}^{G_Y} \sum_{t=1}^{G_{TR}} (I_M(x,y,t) - I_G(x,y,t))^2} \tag{8}$$

where $I_M(x, y, t)$ is the rainfall intensity at the spatial location (*x, y*) at time *t* estimated from crowdsourced observations, and $I_G(x, y, t)$ is the associated ground true rainfall intensity. G_X, G_Y, and G_{TR} are the total number of grid cells in the *X*, *Y*, and time dimensions, respectively.

REAA is a metric for depicting the relative bias in rainfall field estimation:

$$REAA = \frac{\left| \frac{1}{G_X G_Y G_{TR}} \sum_{x=1}^{G_X} \sum_{y=1}^{G_Y} \sum_{t=1}^{G_{TR}} I_M(x,y,t) - \frac{1}{X\,Y\,T_R} \sum_{x=1}^{G_X} \sum_{y=1}^{G_Y} \sum_{t=1}^{G_{TR}} I_G(x,y,t) \right|}{\frac{1}{X\,Y\,N} \sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{n=1}^{N} I_G(x,y,t)} \tag{9}$$

In order to quantify the relative improvement in crowdsourced data quality after quality control (i.e., noise filtering), we introduce the reduction ratio of *REAA* and *RMSE* (i.e., $\Delta REAA$ and $\Delta RMSE$) which are defined as:

$$\Delta REAA = \frac{REAA_n - REAA_f}{REAA_n} \tag{10}$$

$$\Delta RMSE = \frac{RMSE_n - RMSE_f}{RMSE_n} \tag{11}$$

where $REAA_n$ denotes the *REAA* value of crowdsourced rainfall field before noise filtering, and $REAA_f$ is the *REAA* value of crowdsourced rainfall field after noise filtering. The subscripts for *RMSE* share the same definitions as to *REAA*.

3. Results and Discussion

3.1. Error statistics and model performance

We first present CSQC model performances and the spatial distributions of different rainfall fields under the benchmark scenario in San Diego. Figure 3(a) shows the 'ground truth' rainfall field at one representative time-step over a 40 × 20 km^2 space at a 500 m × 500 m resolution. A set of crowdsourced rainfall observations is then generated from the 'ground truth' rainfall (assuming parameters in the benchmark scenario in section 2.3.3) and interpolated (through nearest neighborhood interpolation) into different crowdsourcing rainfall fields [Figure 3(b)-3(d)]. Among them, Figure 3(c) plots the crowdsourcing rainfall field interpolated from the original crowdsourced data with noise, Figure 3(b) plotting a set of data where all noisy observations are correctly removed, and Figure 3(d) plotting the quality controlled crowdsourced data where noisy observations identified by the *k*NN algorithm are filtered. Comparing Figure 3(b) and Figure 3(c), it is found that, during the study time frame, crowdsourcing rainfall field with noise has a higher estimate of rainfall intensity than that without noise. This is consistent with some previous studies as the noisy observations typically overestimate rainfall intensities (Starkey et al., 2017). The quality-controlled crowdsourcing rainfall field in Figure 3(d) also shows lower estimates of rainfall intensity than the crowdsourcing rainfall field with noise. Comparing Figure 3(c) and Figure 3(d), it can be seen that the *k*NN algorithm filters out at least two crowdsourced observations with extremely high rainfall intensity estimates in the center and lower-left part of the study region, which suggests the CSQC model's ability to identify abnormally extreme values.

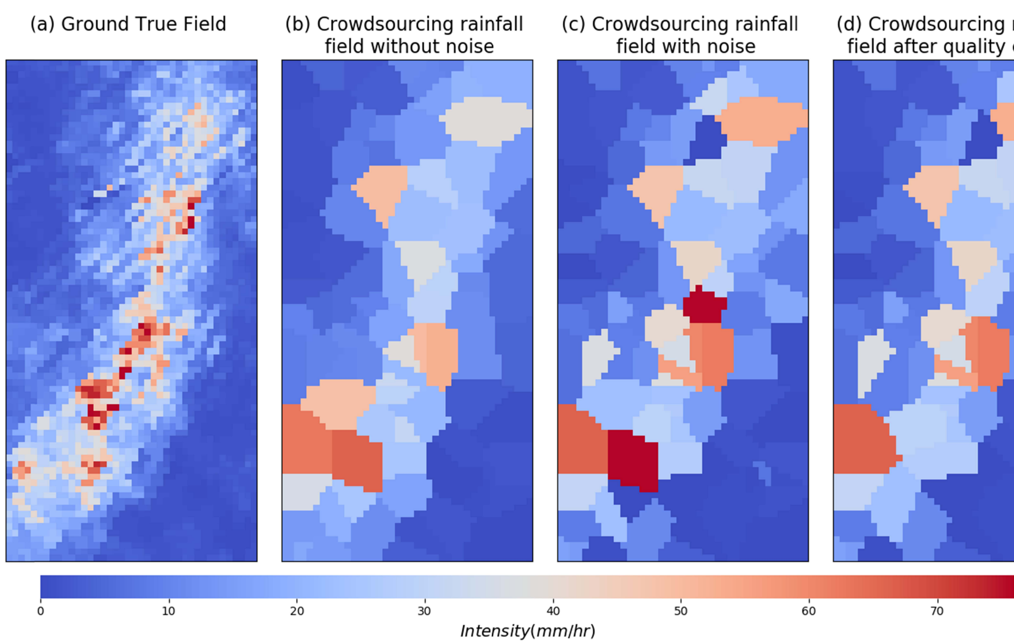

Figure 3. Spatial distribution of rainfall intensity at one representative time-step for (a) the ground true field, the corresponding interpolated rainfall field with (b) crowdsourcing data without noise, (c) crowdsourcing data with noise, and (d) crowdsourcing data after quality control.

We further calculate the errors of the three interpolated crowdsourcing fields (Figure 4). It can be seen that, on average, the crowdsourcing rainfall field with noise [Figure 4(b)] has higher absolute errors than that without noise [Figure 4(a)], especially in the lower-left corner and center region of the study area. Comparing Figure 4(b) and Figure 4(c), it could be seen that the true positive (*TP*) predictions made and removed by the *k*NN algorithm have successfully reduced the high estimation error Figure 4(b), especially at the center region of the study area [Figure 4(c)].

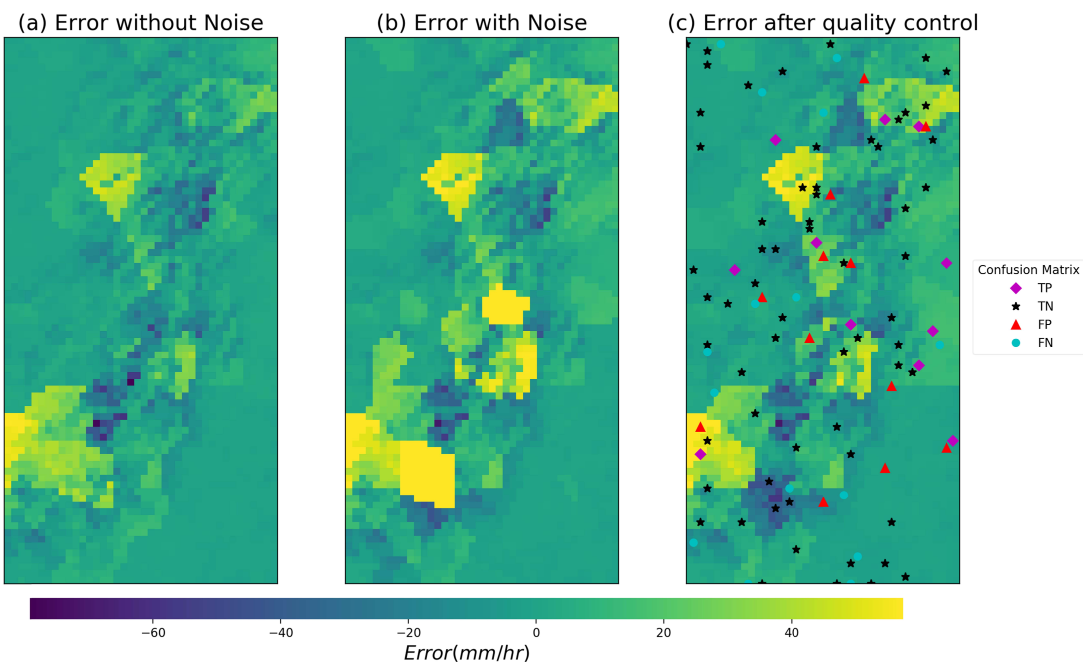

Figure 4. Errors of interpolated rainfall fields at one representative time-step with (a) crowdsourced data without noise, (b) crowdsourced data with noise, and (c) crowdsourced data after quality control; in (c) also shows the labels of true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*) predictions made by the *k*NN model.

A comprehensive comparison of the four machine learning algorithms in section 2.2 for the benchmark scenario is shown in Table 3, where their performances measured through four classification metrics (accuracy scores, *PPV*, *NPV*, and *AUC*) and two rainfall field estimation statistics ($\Delta REAA$ and $\Delta RMSE$) are presented for all the 60 time steps investigated in San Diego. All four machine learning algorithms achieve relatively high accuracy scores over 0.8. The two supervised learning algorithms (*k*NN and MLPs) have higher accuracy scores than unsupervised learning algorithms (iForest and K-Means). MLPs has the highest accuracy score (0.903), followed by *k*NN, K-means, and iForest.

The higher values of NPV over PPV in Table 2 suggest the difficulty to effectively identify less frequent noisy observations in this imbalanced classification task. In general, the two supervised learning algorithms have relatively large PPV values, but the unsupervised K-Means algorithm is the one with the highest PPV value (0.761). The unsupervised iForest algorithm has the lowest PPV value (0.530). MLPs also has the highest AUC value of 0.93, followed by kNN (0.733), iForest (0.546), and K-means (0.534). The rankings of different QCSC algorithms for $\Delta RMSE$ and $\Delta REAA$ are consistent with that of the AUC value. MLPs has the best performance on rainfall field error reduction with $\Delta RMSE$ and $\Delta REAA$ values of 38.10% and 57.68%, respectively. The results indicate that supervised learning algorithms have better performances than unsupervised learning algorithms under the benchmark scenario in San Diego.

Table 3. Quality control results produced from supervised and unsupervised algorithms

	Accuracy	***PPV***	***NPV***	***AUC***	$\Delta RMSE$	$\Delta REAA$
***k*NN**	0.868	0.650	0.897	0.773	32.08 %	53.90 %
MLPs	0.903	0.729	0.933	0.930	38.10 %	57.68 %
iForest	0.832	0.530	0.844	0.546	32.44 %	35.70 %
K-means	0.839	0.761	0.840	0.534	32.00 %	34.55 %

The relatively good accuracy and AUC performances of supervised learning algorithms could be explained by different learning mechanisms between supervised and unsupervised learning algorithms. Compared to unsupervised learning algorithms, supervised learning algorithms inherit extra information from the labels of the training target. While unsupervised learning algorithms are designed to identify the internal patterns in the input features, such internal pattern may or may not coincide with the pattern that is represented by the training labels. Actually, when assigned with the same non-linear task, it is typical that supervised learning algorithms are outperforming the unsupervised learning algorithms, especially when a set of high-quality labels is available (Mohammady et al., 2015; Sathya & Abraham, 2013).

3.2. Sensitivity analysis

It should be noted that, results in section 3.1 show performances of different CSQC algorithms at only the benchmark scenario. As shown in Figure S2 in the SI, the differences in error statistics between the crowdsourcing rainfall fields with and without noise are significantly affected by the assumptions of crowdsourcing noise characteristics, and thus the performances and rankings of different CSQC algorithms vary under different crowdsourcing scenarios (Table 1).

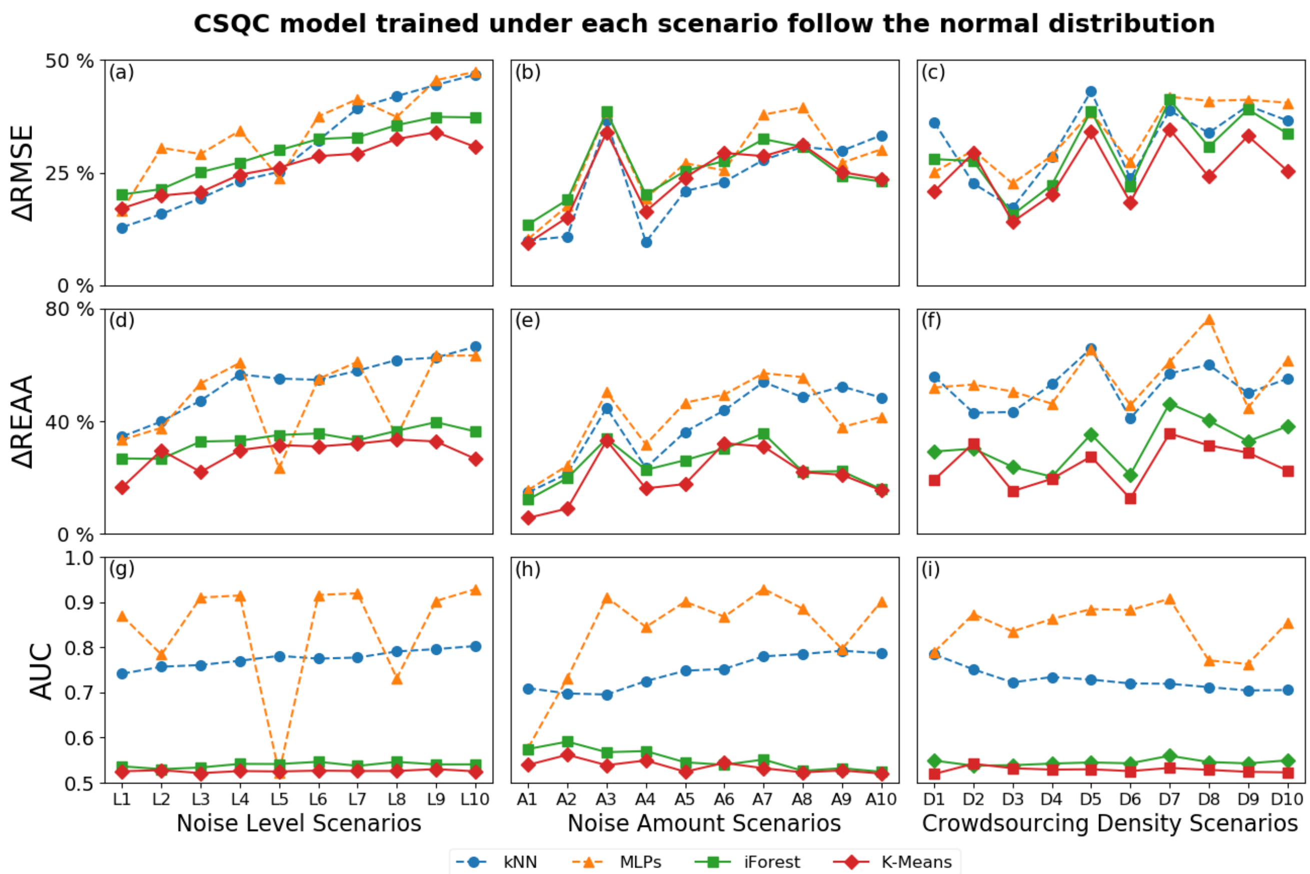

Figure 5. *RMSE* and *REAA* reduction ratio and *AUC* value driven by the four algorithms with noise coefficients follow the normal distribution trained under each scenario in the San Diego, subplots in first row (a-c) presents the $\Delta RMSE$ curve under Noise Level, Noise Amount, and Crowdsourcing Density scenarios, second row (d-f) is $\Delta REAA$ curve under three types scenarios, last row (g-i) is AUC value under three types scenarios.

We thus show the impacts of noise level [Figure 5(a-c)], noise amount [Figure 5(d-f)], and crowdsourcing density [Figure 5(g-i)] on the performances (as measured by $\Delta RMSE$, $\Delta REAA$, and AUC) of the CSQC model driven by the four machine learning algorithms (MLPs, *k*NN, iForest, and K-Means). For each scenario shown in Figure 5, the CSQC models are retrained with synthetic data from that scenario. The result shows a positive impact of noise level on $\Delta RMSE$ for all four machine learning algorithms [Figure 5(a)]. The noise amount and crowdsourcing density also show positive impacts on the $\Delta RMSE$ for all the four machine learning algorithms [Figure 5(b-c)], except for the K-means algorithm which shows no clear trend of $\Delta RMSE$ with the increase of crowdsourcing density [Figure 5(c)]. Overall, MLPs performs at least as good as any other algorithm in terms of $\Delta RMSE$ under all investigated scenarios.

We identify no clear trend in $\Delta REAA$ of the four algorithms with the increase of noise level, noise amount, and crowdsourcing density [Figure 5(d-f)], except for *k*NN and MLPs whose $\Delta REAA$ increase with the increase of noise amount [Figure 5(e)]. When measured with $\Delta REAA$, the two supervised learning algorithms (*k*NN and MLPs) clearly outperform the two unsupervised learning algorithms (iForest and K-means) we have investigated. The advantages of *k*NN and MLPs over iForest and K-means are more obvious when the noise amount is large [e.g., noise amount >30% (A4), Figure 5(e)]. Such result suggests a potential that supervised algorithms might be more robust than the unsupervised algorithms when encountered more anomaly observations.

We also identify no clear trend in AUC values of the four algorithms in Figures 5(g-i). It is shown that supervised learning algorithms obtained higher AUC values than unsupervised learning algorithms under most scenarios, although the AUC value of MLPs has a degree of fluctuation. In general, with crowdsourcing noise following the normal distribution, if the algorithms are retrained under each scenario, the CSQC model driven by supervised learning algorithms achieve better performances than unsupervised learning algorithms for both rainfall field estimation and crowdsourcing noise identification. This is also consistent with model performance under the benchmark scenario in section 3.1.

The trends observed in Figure 5 could be partially explained by results in Figure 6, which shows the $RMSE$ and $REAA$ values of rainfall fields interpolated with noisy crowdsourced data before and after quality control by the MLPs algorithm.

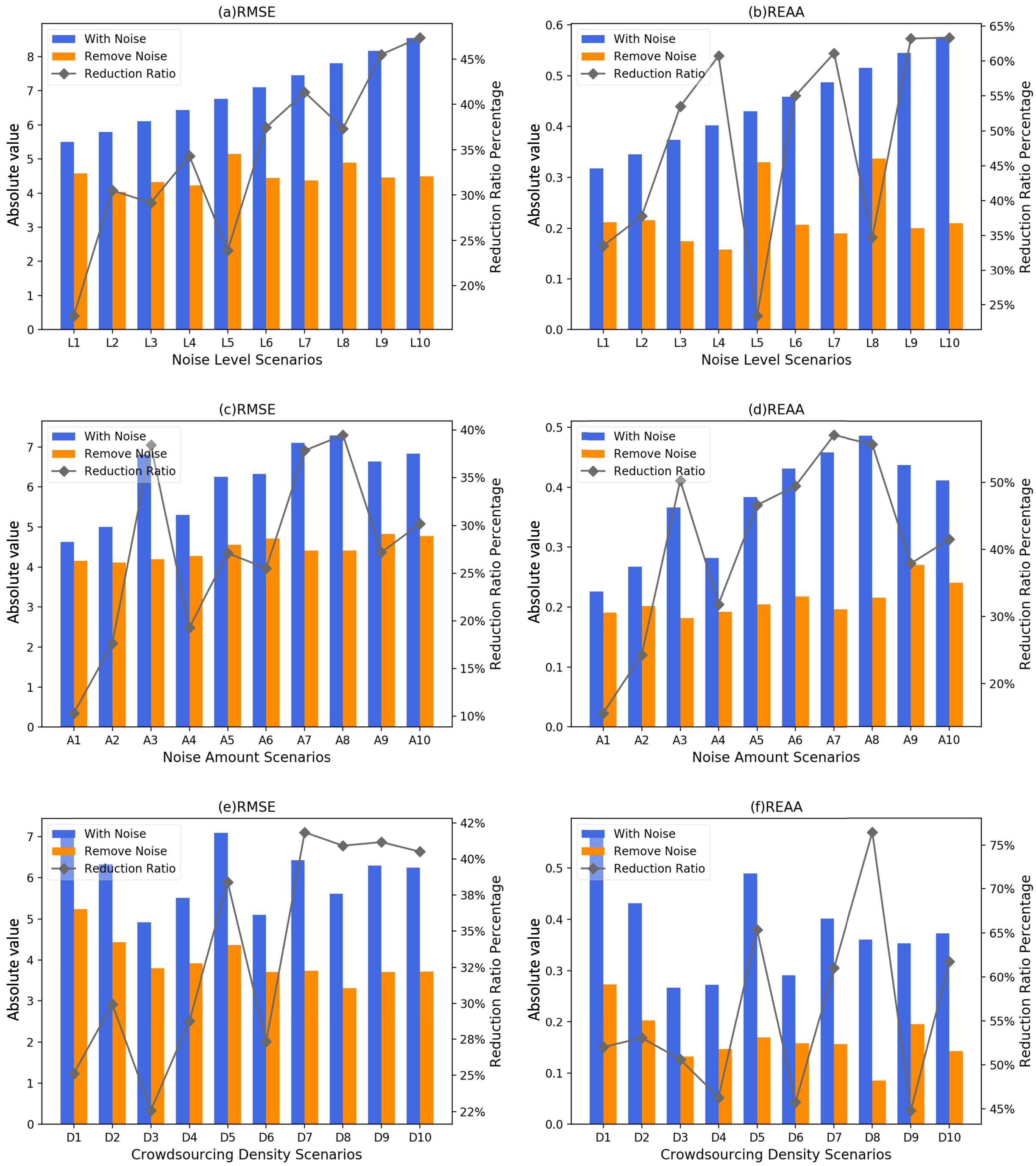

Figure 6. RMSE and REAA values before and after quality control by the MLPs-based CSQC model, values are calculated based on the testing performances of the MLPs algorithm trained under Noise Level (a-b), Noise Amount (c-d), and Crowdsourcing Density (e-f) scenarios in San Diego; refer to Table 1 for details of the scenarios.

For example, with noise coefficients following the normal distribution, the absolute values of *RMSE* and *REAA* before quality control illustrated in Figure 6(a-b) show positive trends with the increase of noise level. However, while the *RMSE* and *REAA* values of the quality controlled crowdsourcing rainfall field fluctuates with the increase of noise level, no clear trend is identified [Figure 6(a-b)]. Similar trends are also true for the impact of noise amount [Figure 6(c-d)]. Such result suggests a relatively stable capability of the MLPs algorithm in identifying noisy observations in the crowdsourcing rainfall data regardless of the statistical characteristics or the composition of noises. Similar conclusion is also true for the supervised learning *k*NN algorithm (Figure S5 in the SI), but not for the two unsupervised learning algorithms. Generally, the crowdsourcing rainfall field quality controlled by iForest and K-Means shows a slight increase of *RMSE* and *REAA* values with noise level, but with a less steep trend than the original crowdsourcing rainfall field without quality control (Figure S6-S7 in the SI).

Because of the randomness in generating crowdsourcing observations, the *RMSE* and *REAA* values before quality control under different crowdsourcing densities [Figure 6(e-f)] show significant fluctuations. In contrast, the *RMSE* and *REAA* values of quality controlled crowdsourcing rainfall field first decrease with the increase of crowdsourcing density, and then reach relatively stable levels after scenario D3 [Figure 6(e-f)]. Such trends are also identified by using other machine learning algorithms shown in Figure S5-S7 in the SI, which suggests the possible contribution of increased number of training data in improving machine learning algorithm performances (Jordan & Mitchell, 2015).

In addition to the analysis of single factor (e.g., noise level or noise amount) impacts on the performances of the CSQC model, Figure 7 presents the interactive impacts of noise level and noise amount on the reduction ratios (*REAA* and *RMSE*) for the CSQC model driven by MLPs. Similar to the findings in Figure 5, the reduction ratios of REAA and RMSE increase with the noise level. Reduction ratios of *REAA* and *RMSE* also show positive correlations with noise amount, though with fluctuations caused by the random crowdsourced observation generation process. The reduction ratios could be as high as 67.05% and 44.30% for *REAA* and *RMSE,* respectively, and are achieved at relatively high levels of noise level and noise amount. We identify no clear higher-order interaction between the noise amount and noise level on the CSQC reduction ratios, i.e., the slope identified for noise level and noise amount in Figure 7 does not vary under our investigated scenarios. Such observation could be partly explained by the relatively stable capability of the CSQC model in identifying noisy crowdsourcing observations, as identified in Figure 6.

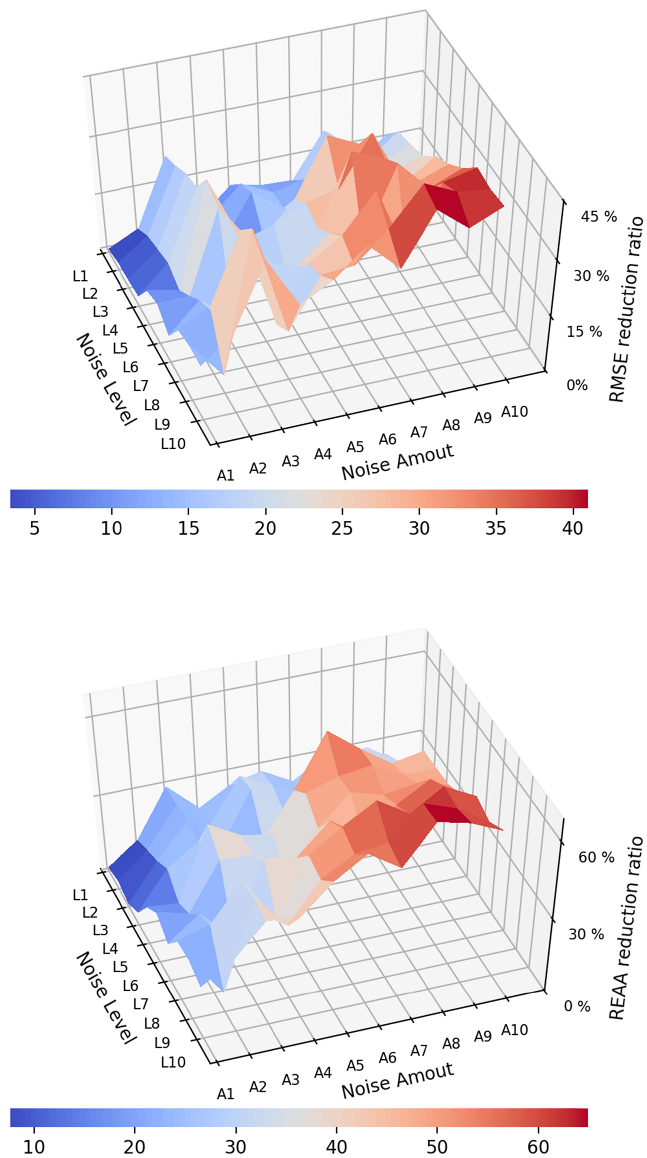

Figure 7. Interactive impacts of noise level and noise amount on the reduction ratios of (a) REAA and (b) RMSE driven by MLPs-based model.

3.3. Model performances under the Wald error distribution

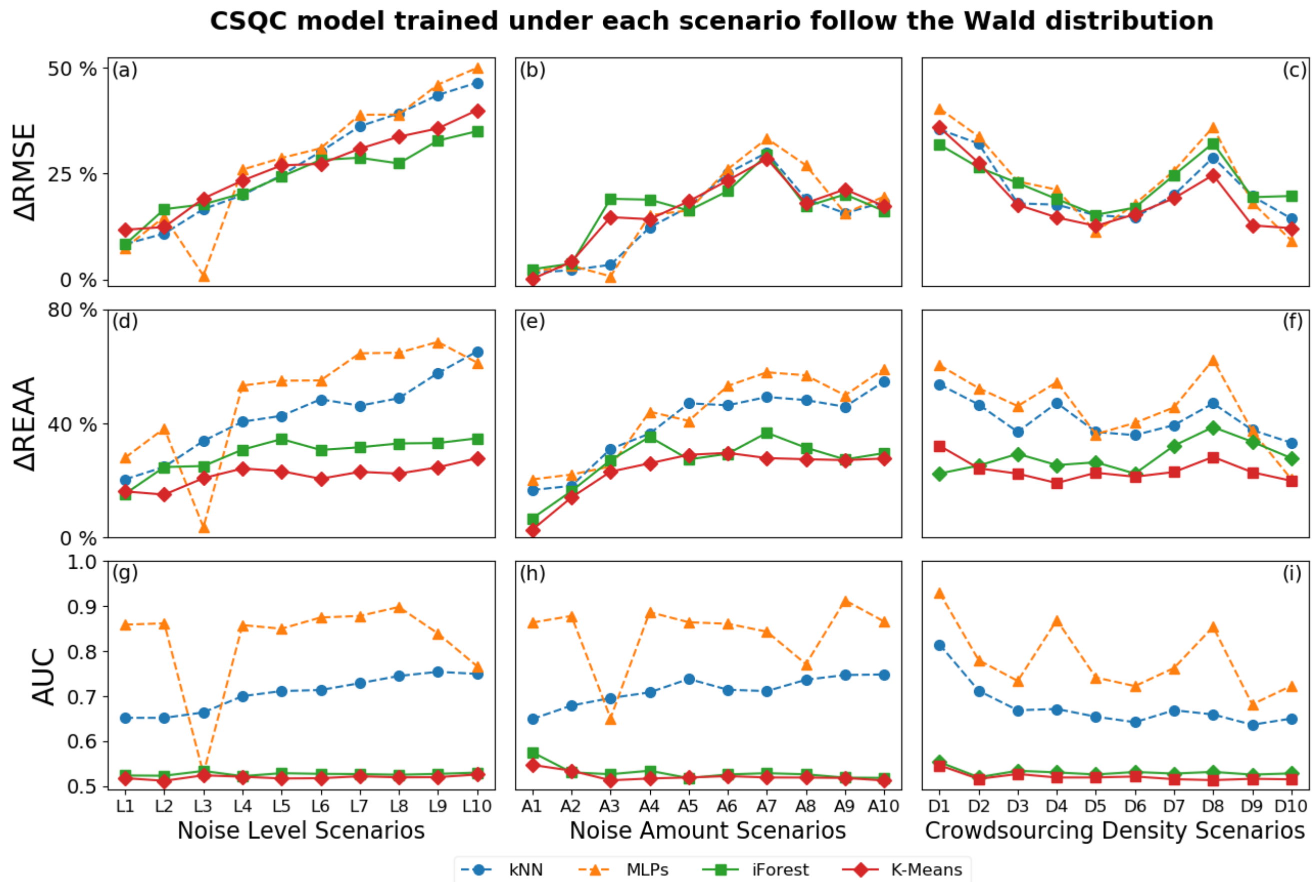

Figure 8. *RMSE* and *REAA* reduction ratio and AUC values driven by the four algorithms with noise coefficients follow the Wald distribution retrained under each scenario in the San Diego, subplots in first row (a-c) presents the $\Delta RMSE$ curve under Noise Level, Noise Amount, and Crowdsourcing Density scenarios, second row (d-f) is $\Delta REAA$ curve under three types scenarios, last row (g-i) is AUC value under three types scenarios.

The performances of the CSQC model might also be affected by the shape of noise distribution in the crowdsourced data. For example, even with the same mean and standard deviation, a positively skewed distribution such as the Wald distribution results in higher frequencies of small noises and lower frequencies of large noises than the normal distribution (see Figure S8 in the SI), and the magnitudes of large noises are also much larger under the Wald distribution. While we do not identify a clear difference between the RMSE and REAA of the noisy crowdsourcing rainfall fields under the assumption of Normal and Wald distributions (Figure S4 in the SI), the distinct error structures in the two types of noise distribution might still lead to different CSQC model performances.

Figure 8 illustrates the *RMSE* and *REAA* reduction ratios and AUC values for the CSQC model, assuming Wald distributed noises. $\Delta RMSE$ under Wald distributed noise shows a positive correlation with noise level [Figure 8(a)]. The $\Delta RMSE$ values of CSQC models follow similar trends shown in Figure 5(a-c). Overall, compared with the case of Normal noise distribution, the differences among the investigated machine learning algorithms are less significant under the Wald noise distribution. MLPs continues to be the best performing algorithm when measured by $\Delta RMSE$.

Unlike Figure 5(d-e), $\Delta REAA$ values of all the investigated machine learning algorithms under Wald noise distribution show a significant increase with noise amount and noise level [Figure 8(d-e)]. As per the comparison of the four machine learning algorithms, similar to Figure 5(d-f), supervised learning algorithms achieve better performances than unsupervised learning algorithms under most scenarios. However, in contrast to our analysis assuming Normal noise distribution, the differences between supervised and unsupervised learning algorithms under Wald noise distribution are less clear. For example, the two supervised learning algorithms in Figure 5(f) (with Normal noise distribution) are consistently outperforming unsupervised learning algorithms under all investigated crowdsourcing density levels, while in Figure 8(f) (with Wald noise distribution) the superiority of supervised learning algorithms over unsupervised learning algorithms is less clear. The unsupervised iForest algorithm even has higher $\Delta REAA$ values than the supervised MLPs under scenarios D10 [Figure 8(f)].

The unsupervised learning algorithms have relatively low *AUC* values ranging from 0.5 to 0.6 [Figure 8(g-i)]. In comparison, the supervised learning algorithms have consistently higher *AUC* values [Figure 8(g-i)], with MLPs as the best performing algorithm under almost all investigated scenarios. Similar to Figure 5(g-i), the AUC values of CSQC models under Wald noise distribution show no clear trend with the increase of noise amount, noise level, and crowdsourcing density [Figure 8(g-i)], except for *k*NN whose *AUC* value increases with the noise level and noise amount as shown in Figure 8(g) and 8(h). On average, the *AUC* values of machine learning algorithms under Normal noise distribution are slightly higher than those under Wald noise distribution.

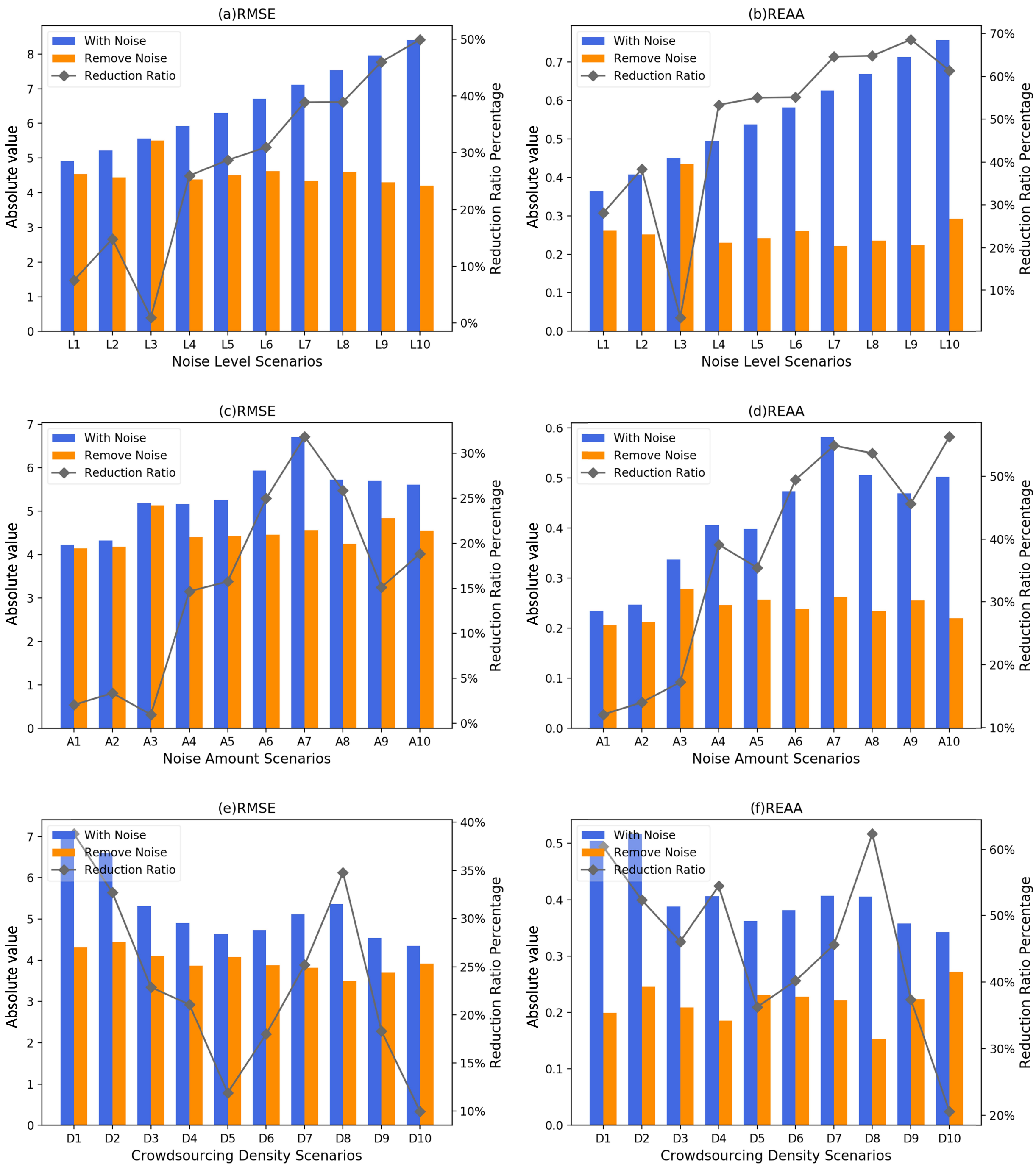

Figure 9. RMSE and REAA values before and after quality control by the MLPs-based CSQC model, values are calculated based on the testing performances of the MLPs algorithm trained under Noise Level (a-b), Noise Amount (c-d), and Crowdsourcing Density (e-f) scenarios in San Diego, where the noise coefficients follow the Wald distribution.

Figure 9 presents the *RMSE* and *REAA* values before and after CSQC model quality control under noise level [Figure 9(a-b)], noise amount [Figure 9(c-d)], and crowdsourcing density [Figure 9(e-f)] scenario, with the noise coefficients following Wald distribution. The result suggests that, similar to the result of crowdsourcing rainfall data with Normally distributed noise in Figure 5, the *RMSE* and *REAA* values of crowdsourcing rainfall field without quality control increase with the noise level and noise amount, and the changes with quality controlled rainfall field are relatively stable [Figure 9(a-d)]. The *RMSE* and *REAA* values of crowdsourcing rainfall field without quality control slightly decrease with the crowdsourcing density, but that of quality controlled rainfall field are relatively stable [Figure 9(e-f)]. Similar results are also true for the *k*NN, iForest, and K-means algorithms (Figures S9-S11 in the SI).

In general, we identify some differences in the CSQC model performances with Normally distributed noise and Wald distributed noise. For example, CSQC model performances with normal noise distribution are more sensitive to changes in noise amount than with Wald noise distribution; the CSQC model achieves a better performance (measured by $\Delta RMSE$, $\Delta REAA$ and AUC) when the noise coefficients follow a Normal distribution; and the difference among the four algorithms are more obvious under the Normal noise distribution. However, the results of the comparison between the four algorithms are overall consistent. Under both the Normal and Wald noise distribution assumptions, supervised learning algorithms (especially the MLPs) are performing better than the unsupervised learning algorithms in both identifying noisy crowdsourced observations and improving the accuracy of quality-controlled crowdsourcing rainfall field.

3.4. Model transferability

In this section, we directly apply the CSQC model trained with the benchmark scenario in San Diego to various scenarios in the three cities without any retraining. The transferability of the CSQC model is then measured by the $\Delta RMSE$, $\Delta REAA$, and AUC values presented in Figures 10-12. Normal noise distribution is assumed for all the analysis in this section. The higher the values of these metrics ($\Delta RMSE$, $\Delta REAA$, and AUC), the better the CSQC model in transferring to other application conditions without further retraining.

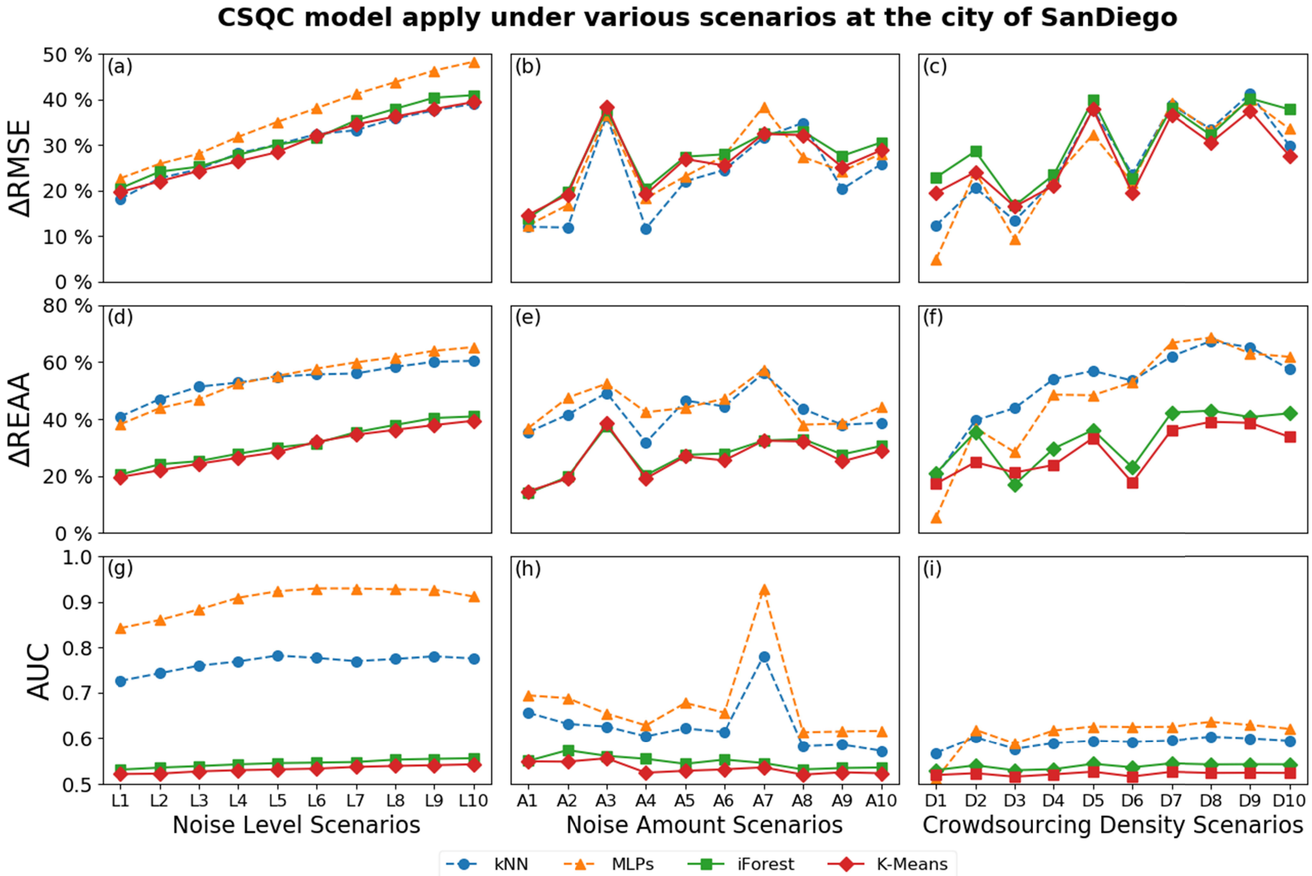

Figure 10. *RMSE* and *REAA* reduction ratio and AUC value by transfer the CSQC model trained under benchmark scenario to other type scenarios in the San Diego, subplots in first row (a-c) presents the $\Delta RMSE$ curve under Noise Level, Noise Amount, and Crowdsourcing Density scenarios, second row (d-f) is $\Delta REAA$ curve under three types scenarios, last row (g-i) is AUC value under three types scenarios.

Figure 10 presents the model transferring performances in San Diego. It is shown that the $\Delta RMSE$, $\Delta REAA$ and AUC values increase monotonically with the noise level [Figure 10 (a), (d), and (g)]; in comparison, when the CSQC model is retrained every time, values of the three metrics also increase with noise level but with much larger variations [Figure 5(a), (d), and (g)]. Comparing the performances of different algorithms, the two supervised learning algorithms, especially MLPs, continue to be the best one in reducing crowdsourcing rainfall field estimation errors and identifying noisy crowdsourced observations, when they are transferred to other scenarios without retraining.

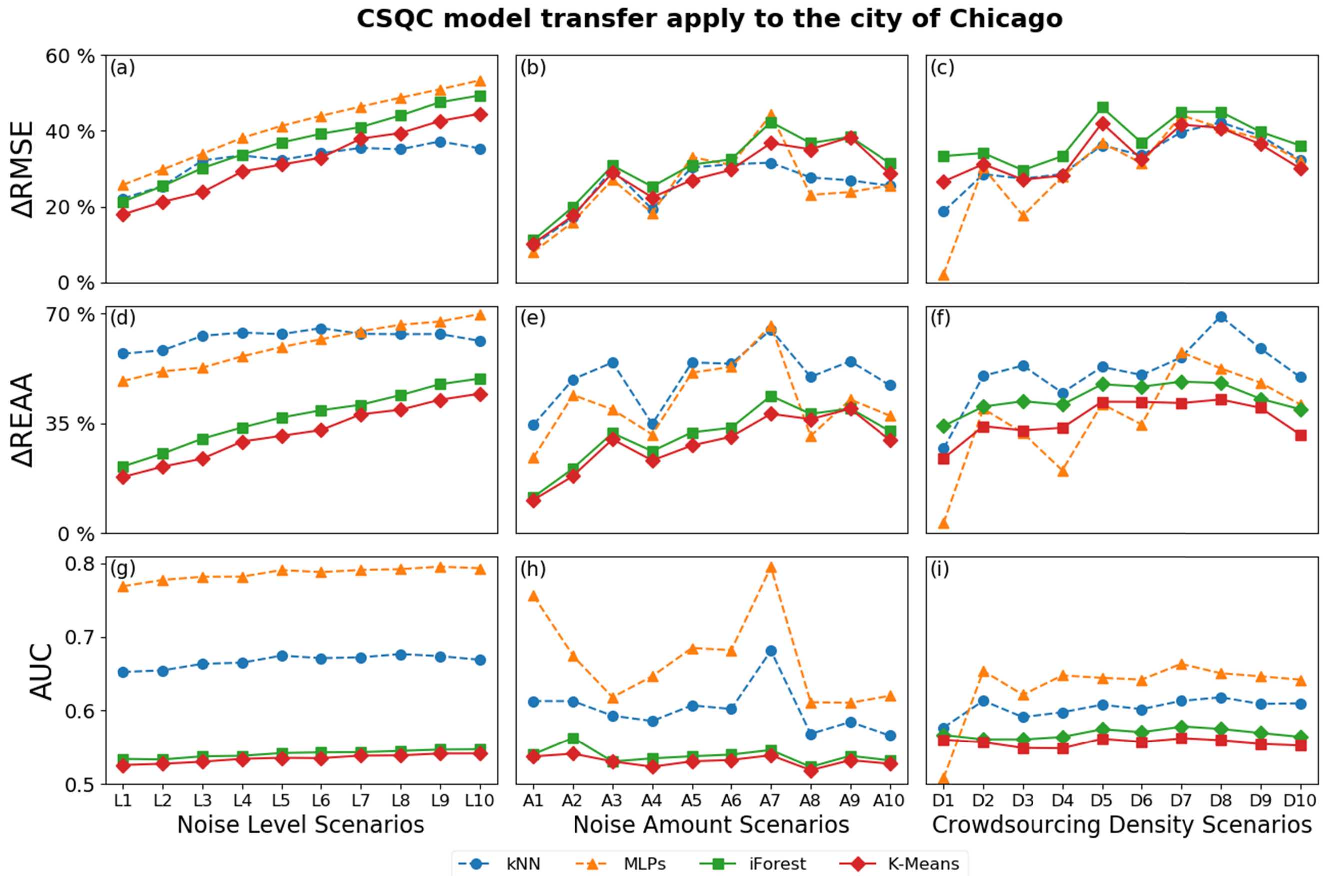

Figure 11. *RMSE* and *REAA* reduction ratio and *AUC* values under three types of scenarios by transfer apply the CSQC model (driven by the four algorithms) to the City of Chicago, subplots in first row (a-c) presents the $\Delta RMSE$ curve under Noise Level, Noise Amount, and Crowdsourcing Density scenarios, second row (d-f) is $\Delta REAA$ curve under three types scenarios, last row (g-i) is AUC value under three types scenarios.

We further explore the CSQC model transferability with rainfall data different from the climate condition in San Diego. More specifically, we directly apply the model trained with the benchmark scenario in San Diego to synthetic data generated from radar observations in the city of Chicago (Figure 11) and Miami (Figure 12).

When the CSQC model is transferred to Chicago, in general, the two supervised learning algorithms (especially the MLPs algorithm) continue to outperform the unsupervised learning algorithms. This is especially true for the AUC performance measure, where MLPs performs the best among all the compared algorithms, followed by *k*NN, K-Means, and iForest [Figure 11(g-i)]. However, compared to the sensitivity analysis with retraining in section 3.2, the relative advantage of the supervised learning algorithms over unsupervised learning algorithms are less significant, and the two unsupervised learning algorithms (iForest and K-Means) even outperform the supervised learning algorithms as shown in Figure 11(c). Similar rankings of the four investigated machine learning algorithms are also identified when the CSQC model is transferred to the city of Miami (Figure 12).

The correlations between the performance measures (i.e., $\Delta RMSE$, $\Delta REAA$, and AUC values) and the noise level, noise amount, and crowdsourcing density, shown in Figures 11-12, respectively, are generally consistent with that identified in Figure 5, i.e., positive correlations for $\Delta RMSE$ and $\Delta REAA$, and no clear trend for AUC. However, compared to the sensitivity analysis results with retraining (Figure 5), the CSQC model shows a lower level of fluctuations under different noise scenarios when it is transferred to Chicago or Miami (Figures 11-12). For example, the $\Delta RMSE$ and $\Delta REAA$ values in Figure 5(c) and Figure 5(f) are not as stable as that in Figure 11(c) and Figure 11(f). Such an elevated level of fluctuation for the retrained CSQC model in Figure 5 is expected. The retraining of a machine learning model typically introduces additional randomness, which increases the fluctuations in the model performances in Figure 5.

Figure 13 displays a summary of the performances of all the four investigated algorithms with and without retraining. Each boxplot in Figure 13 shows the distribution of a performance measure under all the 30 scenarios defined in Table 2. The results suggest that, as expected, the classification accuracies of the two supervised learning algorithms (*k*NN and MLPs) deteriorate when they are not retrained with new test scenario data (T1, T2, and T3 in Figure 13), especially when they are directly applied to a region with the climate significantly different from where they are trained (T2 and T3 in Figure 13). However, we do not identify a clear difference in the performances of the retrained and transferred supervised learning algorithms in reducing crowdsourcing rainfall estimation errors ($\Delta RMSE$ and $\Delta REAA$). Such a difference between the trends of reduction ratios and the AUC values could be possible as the spatial interpolation process for estimating $\Delta RMSE$ and $\Delta REAA$ might smooth out small variations in the filtered crowdsourced observations.

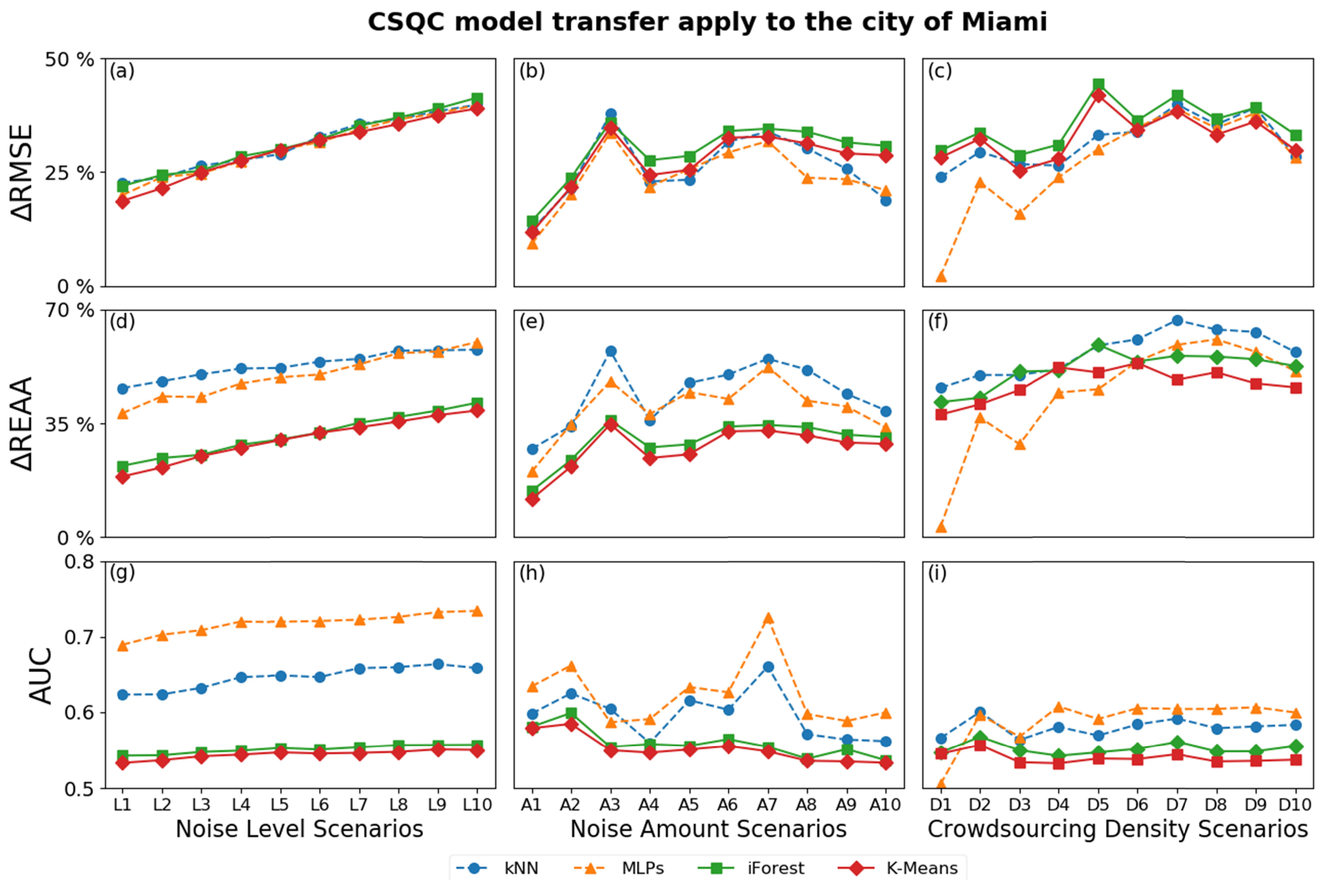

Figure 12. *RMSE* and *REAA* reduction ratio and *AUC* values under three types of scenarios by transfer apply the CSQC model (driven by the four algorithms) to the City of Miami, subplots in first row (a-c) presents the $\Delta RMSE$ curve under Noise Level, Noise Amount, and Crowdsourcing Density scenarios, second row (d-f) is $\Delta REAA$ curve under three types scenarios, last row (g-i) is AUC value under three types scenarios.

Interestingly, all the performance measures of the two unsupervised learning algorithms slightly increase when they are not retrained with new test scenario data (T1, T2, and T3 in Figure 13), especially when the algorithms trained to one city (San Diego) are directly applied to another city (e.g., Chicago) with significantly different climate (T2 and T3 in Figure 13). The results thus suggest a better transferability of unsupervised learning algorithms over supervised learning algorithms. Such an unexpected result can still be understandable as the noisy crowdsourced observations are identified in unsupervised learning algorithms through some internal relationships between the input features. Without the additional constraints brought by the labels in the supervised learning algorithms, it could be possible that an unsupervised learning algorithm trained with one set of data can perform even better with another set of data.

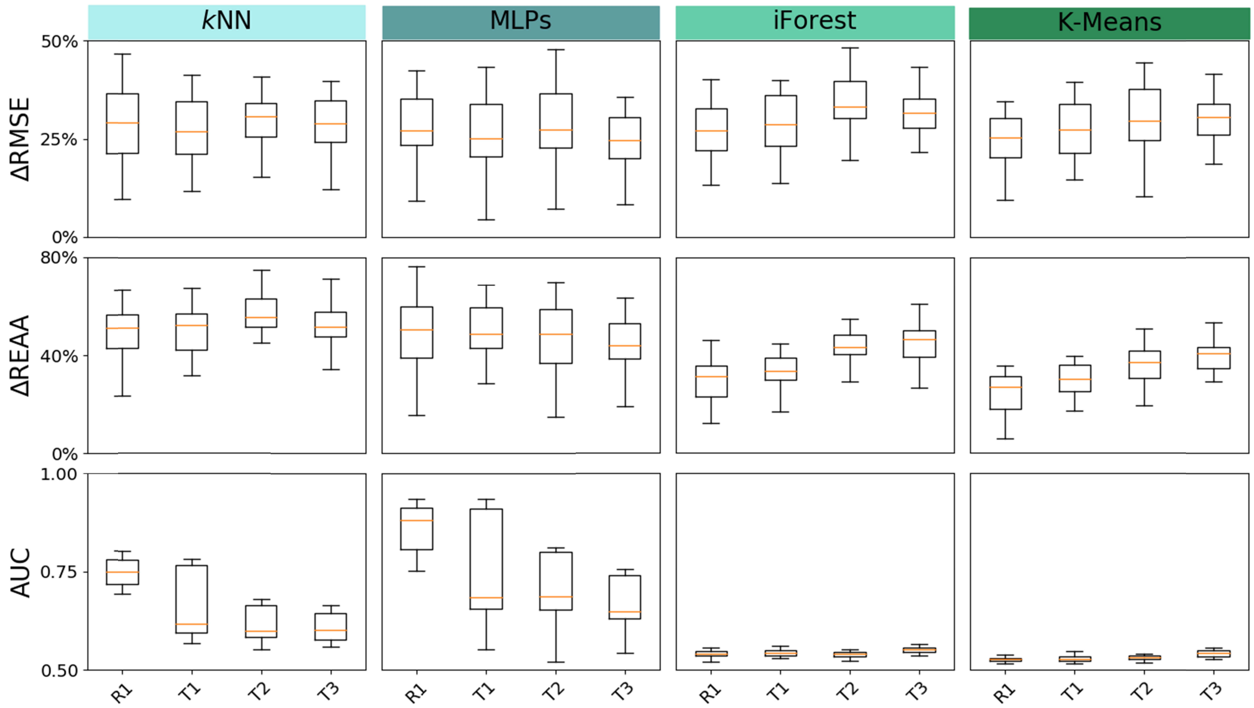

Figure 13. *RMSE* and *REAA* reduction ratio and *AUC* values distribution driven by different machine learning algorithms (R1: Retrain the CSQC model at San Diego; T1: Transfer the trained benchmark model to other scenarios in San Diego; T2: Transfer the benchmark trained model to Chicago; T3: Transfer the benchmark trained model to Miami).

Overall, Figures 10-13 suggest a relatively good transferability of the CSQC model based on either supervised or unsupervised learning algorithms. This is especially true when the CSQC model is applied to reducing the errors in rainfall field estimation (i.e., $\Delta RMSE$ and $\Delta REAA$). The unsupervised learning algorithms possess a better transferability than the supervised learning algorithms, but the supervised learning algorithms (especially MLPs) continue to have better absolute values of the various performance measures ($\Delta RMSE$, $\Delta REAA$, and AUC) when the CSQC model is not retrained.

4. Conclusions

In this study, we propose a machine learning based crowdsourced data quality control (CSQC) method to identify the noisy data in crowdsourced rainfall observations potentially collected from smartphones, personal weather stations, surveillance cameras, and other low-cost devices by citizens. Based on the features extracted from the original crowdsourced data and the interpolated crowdsourcing rainfall field, two supervised learning (MLPs and kNN) and two unsupervised learning (K-means and iForest) algorithms are used to identify noisy observations in the CSQC model. A series of synthetic but realistic scenarios in three cities with different climates are designed to investigate the impacts of the magnitude of noise (noise level), the relative portion of noisy observation (noise amount), and crowdsourcing participation density (crowdsourcing density) on the CSQC model performances. The CSQC model performances are tested and evaluated in terms of their ability to reduce rainfall field estimation errors ($\Delta RMSE$ and $\Delta REAA$) and to identify noisy crowdsourced observations (AUC). Moreover, to test the transferability of the CSQC model, the trained CSQC model in a benchmark scenario in San Diego is directly tested under different scenarios in San Diego, Chicago, and Miami without further retraining.

The four machine learning algorithms (i.e., kNN, MLPs, iForest, and K-means) investigated in this study all show a relatively good performance in reducing the rainfall field estimation errors (i.e., $\Delta RMSE$ and $\Delta REAA$) and identifying noisy crowdsourced observations (i.e., AUC). In general, the two supervised learning algorithms (kNN and MLPs) outperform the two unsupervised learning algorithms (iForest and K-means), and MLPs is the best. The results are consistent across all the testing cases with and without CSQC model retraining, even when the CSQC model trained in one city (San Diego) is directly applied to another city with significantly different rainfall conditions (Chicago or Miami). The results are robust with the various assumptions of noise distribution (i.e., Normal distribution or Wald distribution). More specifically, supervised learning algorithms excel in identifying both noisy and regular observations from a set of crowdsourced data (i.e., PPV and NPV). In contrast, unsupervised learning algorithms can only effectively identify regular observations (i.e., NPV). We find that the noise level positively affects the CSQC model performance measures ($\Delta RMSE$, $\Delta REAA$, and AUC), which is understandable as more distinct noisy observations are easier to identify. The transferability test reveals that, even though the CSQC model performance slightly deteriorates when it is directly applied to a new set of data without retraining, it continues to provide a substantial contribution in rainfall field estimation error reduction and noisy crowdsourced observation identification (Figures 11-13).

Handling concerns over crowdsourced data quality will continue to be a major challenge in the near future (Zheng et al., 2018). While existing quality control methods for crowdsourced observations focus on a special case of fixed-point sensors, our CSQC model is the first to identify and filter noisy points form general crowdsourced observations, which are discontinuous in both time and space. In a real-world setting, crowdsourced observations could come from different sources and regions (e.g., smartphones, CCTV camera, etc.). At this end, the CSQC model is proved to be an effective and robust tool for automatically controlling the data quality with a complex set of data. In addition, the CSQC model can also be used to track the performances of different participants of crowdsourcing projects. Their results could be further used for participant rating or education programs (e.g., feedback from public participate in climate sciences (Pidgeon & Fischhoff, 2011)).

It should be noted that our results are generated with a set of synthetic but realistic rainfall data, and thus the conclusion of this study requires further validation with a large set of real-world crowdsourcing observations (which is currently not available according to the authors' knowledge). Because the scenarios we have investigated largely cover the range of crowdsourced observation errors reported in literature (Mazzoleni et al., 2018; de Vos et al., 2018; de Vos et al., 2019), we expect the major conclusions (e.g., the relative performances of the supervised learning algorithms, and the comparison between CSQC model with and without retraining) hold when such real-world crowdsourcing observations are tested.

Moreover, for the sake of simplicity, our comparison of machine learning algorithms is limited to two supervised learning algorithms (i.e., MLPs and *k*NN) and two unsupervised learning algorithms (i.e., iForest and K-means), with both types are evaluated with a set of data different from the data sets where they are trained (model transferability). However, it is still possible that some other machine learning algorithms can outperform the four investigated algorithms in identifying noisy crowdsourcing observations. Nerveless, the conclusions from this study could provide hints on the choice or design of CSQC machine learning algorithms, i.e., a supervised learning algorithm would be preferred if a set of labeled crowdsourcing data is available, even if such data are coming from places with significantly different rainfall patterns from the target region.

Acknowledgement

This work is supported by the National Science Foundation of China (No. 51961125203). The first author gratefully acknowledges the China Scholarship Council (No. 201806010249). All radar rainfall data used in this study are available in NOAA website (https://www.ncdc.noaa.gov/data-access/radar-data/nexrad) and are cited in this paper. Source code developed to implement the CSQC procedure in this study are available online at (http://doi.org/10.5281/zenodo.4158931).

Reference:

Aggarwal, C. C. (2015). Outlier analysis. In *Data mining* (pp. 237–263). Springer.

Allahbakhsh, M., Benatallah, B., Ignjatovic, A., Motahari-Nezhad, H. R., Bertino, E., & Dustdar, S. (2013). Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, *17*(2), 76–81.

Alpaydin, E. (2014). *Introduction to machine learning*. MIT press.

Altunkaynak, A., & Strom, K. B. (2009). A predictive model for reach morphology classification in mountain streams using multilayer perceptron methods. *Water Resources Research*, *45*(12).

Bauer, P., Mahfouf, J., Olson, W. S., Marzano, F. S., Michele, S. Di, Tassa, A., & Mugnai, A. (2002). Error analysis of TMI rainfall estimates over ocean for variational data assimilation. *Quarterly Journal of the Royal Meteorological Society: A Journal of the Atmospheric Sciences, Applied Meteorology and Physical Oceanography*, *128*(584), 2129–2144.

Bell, S., Cornford, D., & Bastin, L. (2015). How good are citizen weather stations? Addressing a biased opinion. *Weather*, *70*(3), 75–84.

Bhatia, N. (2010). Survey of nearest neighbor techniques. *ArXiv Preprint ArXiv:1007.0085*.

Chen, A. B., Goodall, J. L., Chen, T. D., & Zhang, Z. (2019). The Untapped Potential of Crowdsourced Data for Supporting Flood Resilience: Growing Engagement Faces Non-Uniform Spatial Adoption. In *AGU Fall Meeting 2019*. AGU.

Dennis, B., Ponciano, J. M., Lele, S. R., Taper, M. L., & Staples, D. F. (2006). Estimating density dependence, process noise, and observation error. *Ecological Monographs*, *76*(3), 323–341.

Ding, S., Li, H., Su, C., Yu, J., & Jin, F. (2013). Evolutionary artificial neural networks: a review. *Artificial Intelligence Review*, *39*(3), 251–260.

Durre, I., Squires, M. F., Vose, R. S., Yin, X., Arguez, A., & Applequist, S. (2013). NOAA's 1981–2010 US climate normals: Monthly precipitation, snowfall, and snow depth. *Journal of Applied Meteorology and Climatology*, *52*(11), 2377–2395.

Ebert, T., Eichstaedt, J. C., Lee, N., Obschonka, M., & Rodríguez-Pose, A. (2018). *Big Data, artificial intelligence and the geography of entrepreneurship in the United States*. CEPR Discussion Papers.

Fan, X., Liu, J., Wang, Z., & Jiang, Y. (2016). Navigating the last mile with crowdsourced driving information. In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 346–351). IEEE.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Fencl, M., Dohnal, M., Rieckermann, J., & Bareš, V. (2017). Gauge-adjusted rainfall estimates from commercial microwave links. *Hydrology and Earth System Sciences*, *21*(1), 617–634.

Foody, G. M., See, L., Fritz, S., Van der Velde, M., Perger, C., Schill, C., & Boyd, D. S. (2013). Assessing the accuracy of volunteered geographic information arising from multiple contributors to an internet based collaborative project. *Transactions in GIS*, *17*(6), 847–860.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42.

Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS One*, *11*(4), e0152173.

Goodchild, M. F., & Li, L. (2012). Assuring the quality of volunteered geographic information. *Spatial Statistics*, *1*, 110–120.

Gosset, M., Kunstmann, H., Zougmore, F., Cazenave, F., Leijnse, H., Uijlenhoet, R., et al. (2016). Improving rainfall measurement in gauge poor regions thanks to mobile telecommunication networks. *Bulletin of the American Meteorological Society*, *97*(3), ES49–ES51.

Guo, H., Huang, H., Sun, Y. E., Zhang, Y., Chen, S., & Huang, L. (2019). Chaac: Real-Time and Fine-Grained Rain Detection and Measurement Using Smartphones. *IEEE Internet of Things Journal*, *6*(1), 997–1009. https://doi.org/10.1109/JIOT.2018.2866690

Haklay, M. (2013). Citizen science and volunteered geographic information: Overview and typology of participation. In *Crowdsourcing geographic knowledge* (pp. 105–122). Springer.

Hunt, V. M., Fant, J. B., Steger, L., Hartzog, P. E., Lonsdorf, E. V, Jacobi, S. K., & Larkin, D. J. (2017). PhragNet: crowdsourcing to investigate ecology and management of invasive Phragmites australis (common reed) in North America. *Wetlands Ecology and Management*, *25*(5), 607–618.

Jiang, S., Babovic, V., Zheng, Y., & Xiong, J. (2019). Advancing Opportunistic Sensing in Hydrology: A Novel Approach to Measuring Rainfall With Ordinary Surveillance Cameras.

Water Resources Research, *55*(4), 3004–3027. https://doi.org/10.1029/2018WR024480

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 881–892.

Kazai, G., Kamps, J., & Milic-Frayling, N. (2013). An analysis of human factors and label accuracy in crowdsourcing relevance judgments. *Information Retrieval*, *16*(2), 138–178.

Lease, M. (2011). On quality control and machine learning in crowdsourcing. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Leigh, C., Alsibai, O., Hyndman, R. J., Kandanaarachchi, S., King, O. C., McGree, J. M., et al. (2019). A framework for automated anomaly detection in high frequency water-quality data from in situ sensors. *Science of The Total Environment*, *664*, 885–898.

Lima, M. F., Zarpelao, B. B., Sampaio, L. D. H., Rodrigues, J. J. P. C., Abrao, T., & Proença, M. L. (2010). Anomaly detection using baseline and k-means clustering. In *SoftCOM 2010, 18th International Conference on Software, Telecommunications and Computer Networks* (pp. 305–309). IEEE.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422). IEEE.

Mazzoleni, M., Verlaan, M., Alfonso, L., Monego, M., Norbiato, D., Ferri, M., & Solomatine, D. P. (2017). Can assimilation of crowdsourced data in hydrological modelling improve flood prediction? *Hydrology and Earth System Sciences*, *21*(2), 839–861.

Mazzoleni, M., Juliette Cortes Arevalo, V., Wehn, U., Alfonso, L., Norbiato, D., Monego, M., et al. (2018). Exploring the influence of citizen involvement on the assimilation of crowdsourced observations: a modelling study based on the 2013 flood event in the Bacchiglione catchment (Italy). *Hydrology and Earth System Sciences*, *22*(1), 391–416.

Meier, F., Fenner, D., Grassmann, T., Otto, M., & Scherer, D. (2017). Crowdsourcing air temperature from citizen weather stations for urban climate research. *Urban Climate*, *19*, 170–191.

Middleton, S. E., Middleton, L., & Modafferi, S. (2013). Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems*, *29*(2), 9–17.

Moatar, F., Fessant, F., & Poirel, A. (1999). pH modelling by neural networks. Application of control and validation data series in the Middle Loire river. *Ecological Modelling*, *120*(2–3), 141–156.

Mohammady, M., Moradi, H. R., Zeinivand, H., & Temme, A. (2015). A comparison of supervised, unsupervised and synthetic land use classification methods in the north of Iran. *International Journal of Environmental Science and Technology*, *12*(5), 1515–1526.

NOAA. (2013). NOAA National Weather Service (NWS) Radar Operations Center (1991): NOAA Next Generation Radar (NEXRAD) Level 2 Base Data. *NOAA National Centers for Environmental Information*. https://doi.org/10.7289/V5W9574V

Overeem, A., Leijnse, H., & Uijlenhoet, R. (2016). Retrieval algorithm for rainfall mapping from microwave links in a cellular communication network. *Atmospheric Measurement Techniques*, *9*(5), 2425–2444.

Palen, L., & Anderson, K. M. (2016). Crisis informatics—New data for extraordinary times. *Science*, *353*(6296), 224–225.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, *4*(2), 1883.

Pidgeon, N., & Fischhoff, B. (2011). The role of social and decision sciences in communicating uncertain climate risks. *Nature Climate Change*, *1*(1), 35–41.

Pingali, S. (2017). Cloud Computing and Crowdsourcing for Monitoring Lakes in Developing Countries. *Proceedings - 2016 IEEE International Conference on Cloud Computing in Emerging Markets, CCEM 2016*, 161–163. https://doi.org/10.1109/CCEM.2016.037

Rabiei, E., Haberlandt, U., Sester, M., Fitzner, D., & Wallner, M. (2016). Areal rainfall estimation using moving cars–computer experiments including hydrological modeling. *Hydrology and Earth System Sciences*, *20*(9), 3907–3922.

Ranawana, R., & Palade, V. (2006). Optimized precision-a new measure for classifier performance evaluation. In *2006 IEEE International Conference on Evolutionary Computation* (pp. 2254–2261). IEEE.

Sahoo, S., Russo, T. A., Elliott, J., & Foster, I. (2017). Machine learning algorithms for modeling groundwater level changes in agricultural regions of the US. *Water Resources Research*, *53*(5), 3878–3895.

Saini, I., Singh, D., & Khosla, A. (2013). QRS detection using K-Nearest Neighbor algorithm (KNN) and evaluation on standard ECG databases. *Journal of Advanced Research*, *4*(4), 331–344.

Sathya, R., & Abraham, A. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, *2*(2), 34–38.

Schneider, P., Castell, N., Vogt, M., Dauge, F. R., Lahoz, W. A., & Bartonova, A. (2017). Mapping urban air quality in near real-time using observations from low-cost sensors and model information. *Environment International*, *106*, 234–247.

Starkey, E., Parkin, G., Birkinshaw, S., Large, A., Quinn, P., & Gibson, C. (2017). Demonstrating the value of community-based ('citizen science') observations for catchment modelling and characterisation. *Journal of Hydrology*, *548*, 801–817.

Steger, C., Butt, B., & Hooten, M. B. (2017). Safari Science: assessing the reliability of citizen science data for wildlife surveys. *Journal of Applied Ecology*, *54*(6), 2053–2062.

Talagala, P. D., Hyndman, R. J., Leigh, C., Mengersen, K., & Smith-Miles, K. (2019). A Feature-Based Procedure for Detecting Technical Outliers in Water-Quality Data From In Situ Sensors. *Water Resources Research*. https://doi.org/10.1029/2019WR024906

de Vos, L.W., Droste, A. M., Zander, M. J., Overeem, A., Leijnse, H., Heusinkveld, B. G., et al. (2019). Hydrometeorological monitoring using opportunistic sensing networks in the Amsterdam metropolitan area. *Bulletin of the American Meteorological Society*, BAMS-D-19-0091.1. https://doi.org/10.1175/BAMS-D-19-0091.1

De Vos, L W, Raupach, T. H., Leijnse, H., Overeem, A., Berne, A., & Uijlenhoet, R. (2018). High-resolution simulation study exploring the potential of radars, crowdsourced personal weather stations, and commercial microwave links to monitor small-scale urban rainfall. *Water Resources Research*, *54*(12), 10–293.

de Vos, Lotte Wilhelmina, Leijnse, H., Overeem, A., & Uijlenhoet, R. (2019). Quality Control for Crowdsourced Personal Weather Stations to Enable Operational Rainfall Monitoring. *Geophysical Research Letters*, *46*(15), 8820–8829.

Walker, D., Forsythe, N., Parkin, G., & Gowing, J. (2016). Filling the observational void: Scientific value and quantitative validation of hydrometeorological data from a community-based monitoring programme. *Journal of Hydrology*, *538*, 713–725.

Wu, W., & Wang, Y. (2019). *The Geography of Mobility, Wellbeing and Development in China: Understanding Transformations Through Big Data*. Routledge.

Yang, P., & Ng, T. L. (2017). Gauging through the crowd: A crowd-sourcing approach to urban rainfall measurement and storm water modeling implications. *Water Resources Research*, *53*(11), 9462–9478.

Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017). Efficient knn classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(5), 1774–1785.

Zheng, F., Tao, R., Maier, H. R., See, L., Savic, D., Zhang, T., et al. (2018). Crowdsourcing Methods for Data Collection in Geophysics: State of the Art, Issues, and Future Directions. *Reviews of Geophysics*, *56*(4), 698–740. https://doi.org/10.1029/2018RG000616