5-1-2023

Data-Driven Air Quality and Environmental Evaluation for Cattle Farms

Jennifer Hu
University of California, Berkeley

Rushikesh Jagtap
San Jose State University

Rishikumar Ravichandran
San Jose State University

Chitra Priyaa Sathya Moorthy
San Jose State University

Nataliya Sobol
San Jose State University

See next page for additional authors

Recommended Citation

Jennifer Hu, Rushikesh Jagtap, Rishikumar Ravichandran, Chitra Priyaa Sathya Moorthy, Nataliya Sobol, Jane Wu, and Jerry Gao. "Data-Driven Air Quality and Environmental Evaluation for Cattle Farms" *Atmosphere* (2023). https://doi.org/10.3390/atmos14050771

atmosphere

Article

Data-Driven Air Quality and Environmental Evaluation for Cattle Farms

Jennifer Hu [1,2], Rushikesh Jagtap [2], Rishikumar Ravichandran [2], Chitra Priyaa Sathya Moorthy [2], Nataliya Sobol [2], Jane Wu [3] and Jerry Gao [2,*]

1 Division of Computing, Data Science, and Society, University of California, Berkeley, CA 94720, USA
2 Department of Applied Data Science, San Jose State University, San Jose, CA 95192, USA
3 BRI, San Francisco, CA 94104, USA
* Correspondence: jerry.gao@sjsu.edu

Abstract: The expansion of agricultural practices and the raising of animals are key contributors to air pollution. Cattle farms contain hazardous gases, so we developed a cattle farm air pollution analyzer to count the number of cattle and provide comprehensive statistics on different air pollutant concentrations based on severity over various time periods. The modeling was performed in two parts: the first stage focused on object detection using satellite data of farm images to identify and count the number of cattle; the second stage predicted the next hour air pollutant concentration of the seven cattle farm air pollutants considered. The output from the second stage was then visualized based on severity, and analytics were performed on the historical data. The visualization illustrates the relationship between cattle count and air pollutants, an important factor for analyzing the pollutant concentration trend. We proposed the models Detectron2, YOLOv4, RetinaNet, and YOLOv5 for the first stage, and LSTM (single/multi lag), CNN-LSTM, and Bi-LSTM for the second stage. YOLOv5 performed best in stage one with an average precision of 0.916 and recall of 0.912, with the average precision and recall for all models being above 0.87. For stage two, CNN-LSTM performed well with an MAE of 3.511 and an MAPE of 0.016, while a stacked model had an MAE of 5.010 and an MAPE of 0.023.

Keywords: cattle farm; object detection; regression; air pollution; livestock

check for updates

Citation: Hu, J.; Jagtap, R.; Ravichandran, R.; Sathya Moorthy, C.P.; Sobol, N.; Wu, J.; Gao, J. Data-Driven Air Quality and Environmental Evaluation for Cattle Farms. *Atmosphere* **2023**, *14*, 771. https://doi.org/10.3390/atmos14050771

Academic Editor: Pavel Kishcha

Received: 19 March 2023
Revised: 8 April 2023
Accepted: 21 April 2023
Published: 23 April 2023

1. Introduction

The world population has been expanding exponentially, with this trend expecting to continue in the years to come [1]. In order to meet the growing demand for food and other necessities, it is essential to increase the available supply of these products [2]. Cattle farming is one such contributor to the increasing supply of people's essentials for meat, dairy, and other cattle products, and it has a substantial impact on multiple global environmental issues including climate change and agricultural emissions [3]. The increasing number of farms and livestock contributes considerably to the pollution of the air we breathe [4]. Hazardous gases, such as carbon monoxide (CO), nitric oxide (NO), nitrogen dioxide (NO_2), methane (CH_4), carbon dioxide (CO_2), particulate matter ($PM_{2.5}$, PM_{10}), and ammonia (NH_3), are among the gases found in the troposphere on cow farms, according to researchers [5]. The dangers of air pollution for human health are now widely recognized by the scientific community. According to the American Journal of Respiratory and Critical Care Medicine, individuals who live on or near cattle farms are at significant risk of developing chronic diseases [6]. Every year in the United States, it is estimated that air pollution from livestock farms causes approximately 17,900 deaths [7]. Cattle farms have significant challenges in serving the needs of an expanding and changing consumer base while simultaneously taking measures to decrease their negative impact on air pollution and other environmental problems. CH_4 and N_2O are primary greenhouse gas emissions from livestock, respectively accounting for 44% and 29% of total industry emissions [8].

Due to the severity of this situation, there is a need for creating a detailed analysis of air quality in relation to the cattle count of farms and nearby areas.

The research presented in this paper aimed to keep the area safe by ensuring that livestock farms maintain adequate air quality. The collection, monitoring, and analysis of data on the impact of cattle farms on air quality and environmental effects are necessary to understand the situation better and take appropriate actions [9]. It can also help food firms produce healthier food methods that release less hazardous emissions [10]. With the help of remote sensing data, the severity of each of the air contaminants was estimated and plotted on a map for specific geographic areas.

There are a variety of tools and methodologies available for real-time monitoring of greenhouse gas emissions from cattle farms. Various models found in previous studies for predicting pollutant concentrations include statistical, numerical, and machine learning methodologies such as XGBoost, Random Forest, and deep learning [8,11–13]. For this paper, a system was created for finding the number of cattle, pollutant prediction, classification, and visualization of the severity of pollutants. The first problem was determined as an object detection problem, the second was a prediction problem and the third was a classification and visualization problem. Specifically, data from API [14], hourly remote sensing (the level of pollutants), satellite data, farm data, and location were all incorporated to train the model for air quality monitoring, while satellite images were used to find the cattle count. By using remote sensing data from areas that are not farms, the system can learn how to read different input values and help to understand how much pollution is in the cattle fields. We designed a two-stage supervised model for regression and then performed the classification task. In addition, we developed a web portal with two models to estimate the ratio of each pollutant in the environment and classified them according to the severity of their impact. The classification model for presenting the live severity of the selected farms includes a map visualization that indicates the AQI [15] of greenhouse gases. In addition, we provided a classification method for predicting the air quality of livestock farms and analytics based on the number of cattle present. Farm owners will obtain an understanding through map visualization of the amount of pollution the farm extension activity can cause. They can use the model's output to either lower the number of animals on the farm or increase the size of the field in order to reduce the overall pollution level. This study introduced a novel deep learning approach to analyzing air pollution on cattle farms by combining cattle counting using satellite imaging and air quality prediction to analyze the correlation between cattle population and air pollution. In addition, the system developed in this study combines cattle detection, AQI classification, pollutant prediction, and a map visualization application compared to other studies that only address one or two of these components.

The rest of this paper is structured as follows: Section 2 reviews the related technology and literature; Section 3 describes the data collection and preparation; Section 4 details the design and development of each proposed model; Section 5 summarizes the system development and structure; Section 6 concludes the paper and discusses future work.

2. Related Work

2.1. Technology and Solution Survey

There are a variety of different technologies and solutions that can be used for our project. However, for a better understanding of the different models available and to be able to choose the best among them, we made a comparison between all of the latest algorithms we have found in Table 1.

Table 1. Technologies and Solutions.

Model	Purpose	Advantages	Disadvantages
SVR [16]	Predicts discrete values closest to the hyperplane within a threshold value	1. Easy to update. 2. Considers points with least error rate. 3. Better model fitting. 4. Requires less computational power.	1. Less accurate for noisy data. 2. Not suitable for large datasets. 3. Underperforms when number of features is greater than training sample.
Naive Bayes [17]	Calculates probability based on naive independence assumptions for real-time predictions	1. Highly scalable and performs better than multi-class classifications. 2. Suitable for continuous and discrete data.	1. Unable to learn variable relationships. 2. Assumes all features are independent. 3. Assumes zero frequency problem.
KNN [18]	Classifies data points based on proximity to each other	1. Suitable for noisy and large training data. 2. No training time required. 3. New data can be updated in real time.	1. Not suitable for higher dimensional data. 2. Has a high computing cost.
U-Net CNN [19]	Inputs images and outputs a label for biomedical image segmentation	1. Allows for pixel-based classification. 2. Can work on smaller datasets.	1. Requires data to be labeled. 2. Requires features to be labeled, which needs domain knowledge.
(Linear) LRM [20]	Evaluates trends by assuming linear relationship between input and output variables	1. Works best with linearly separable data. 2. Suitable for smaller number of parameters.	1. Prone to overfitting. 2. Not suitable for noisy data.
YOLO [21]	Predicts objects in real time by splitting input image into grids to generate bounding boxes	1. Model is open source. 2. Faster and more accurate image classification than other models.	1. Grid cells can only predict a single class at a time. 2. Spatial constraints are strong.
RF [22]	Predicts and classifies data based on randomly created decision trees	1. Can be used on datasets with missing values. 2. More accurate than most classification algorithms.	1. Provides less data interpretation. 2. More difficult to visualize. 3. Has a high memory cost for datasets with many features.
XGBoost [23]	Classifies large datasets using gradient boosting framework with parallel decision trees boosting	1. Deals with missing values. 2. Has several useful hyperparameters.	1. Sensitive to outliers. 2. Not suitable for structured data.
VGG-Net [24]	Recognizes images using VGG, 2D convolution, and max pooling	1. Has benchmarking for tasks. 2. Model is open source and free.	1. Difficult to train. 2. Network architecture has large weights.
(Logistic) LR [25]	Predicts pollutant concentrations with discrete outcome probabilities	1. Can be scaled to multiple classes. 2. Classifies unknowns quickly.	1. Has linear boundaries. 2. Requires many features.
LSTM [26]	Predicts pollutant concentrations by selectively remembering patterns from historical data	1. Provides a wide range of parameters for tuning. 2. Handles the vanishing gradient problem.	1. Has a high computing and memory cost for real-world problems. 2. Prone to overfitting. 3. Requires large datasets.

Table 1. *Cont.*

Model	Purpose	Advantages	Disadvantages
RetinaNet [27]	Detects objects in satellite images using single-state object detection	1. Better performance than traditional two stage models. 2. Suitable for dense object detection.	1. Generates a large number of negative samples. 2. Slower detection speed than YOLO.

The above models include machine learning and deep learning techniques, which were used in previous studies for the classification of pollutants, the detection of cattle, image segmentation, predictions of the gases emitted, and cattle counting. A combination of these models also provided a good result for the current study. These models overcome the weaknesses of a single model and perform a given task with better accuracy and performance. Random Forest, KNN, and linear regression are the models mainly used for the classification of pollutants and their severity, whereas models such as YOLO, VGG, and U-Net CNN and RetinaNet are used for detecting cattle from remote sensing and satellite images.

2.2. Literature Survey

Depending on the requirements of this study, the papers related to remote sensing data were helpful in understanding the handling of the captured real-time air pollutant concentration. As part of the literature survey, various research works and journals were parsed to ascertain an idea of how to go about the cattle farm air quality analysis project [28–41]. The search proceeded across four different categories of models using the following data:

1. Remote sensing data;
2. Satellite dataset;
3. Air pollution data;
4. Cattle farm data.

Table 2 shows the comparison of papers that used remote sensing data for air pollution prediction models. Monthly AOD and elevation were the most important predictors in the model, according to variable importance analysis [29]. Cubist, Random Forest, and eXtreme Gradient Boosting were the algorithms that performed the best, according to the results, with Cubist being the best [5]. The authors of [30] suggest using Radial Basis Function (RBF) with Support Vector Regression for greater accuracy. The table shows the comparison of projects that developed prediction models for air quality analysis on cattle farms. The authors of [33] used eight different regression approaches with evaluation metrics of four accuracy measures, and used correlation analysis to select the significant features. To forecast hourly PM_{10} concentrations, this research used the Gradient Boosted Regression Trees (GBRT) model trained to forecast them [38]. Through this method, the effects of diverse environmental variables on PM_{10} can be isolated and quantified. In general, the model is more sensitive to climatic conditions than it is to land cover features. In 2020, an air quality forecasting model was created using satellite data using deep neural networks [39]. The authors used CNN, RNN, LSTM, and a spatiotemporal deep network. The results suggest that using the CNN-LSTM model with the satellite data gives a lower RMSE and MAE.

Papers on working with satellite imagery data were explored to handle the counting mechanism used by different researchers for counting objects such as cars and buildings to help us with cattle counting on each farm. Table 3 shows the comparison of projects that have implemented counting algorithms for modeling. In [39], Google Earth API was used as the data source and a framework was built that collects satellite data of different regions using the attention based re-weighting technique. FusionNet was used to capture the build structure density and MAE and R-squared measures were used to find the performance.

Some studies used deep learning models such as CNN, CSRNet and LCFCN for cattle detection and counting [39–42].

Table 2. Literature survey of air pollution prediction methods.

Reference	Region	Purpose	Model	Metrics	Input Parameters
[28]	Canada	Predicting ground level $PM_{2.5}$	MLR BRNN SVM LASSO MARS RF	CV-RMSE CV-R^2	(Data: MODIS) $PM_{2.5}$, AOD, LST, NDVI, HPBL, wind speed, elevation, distance, month
[29]	Andhra Pradesh, India	Predicting air pollution	SVR MVR	RMSE	(Data: LANDSAT ETM+, IRS P6) NDVI, TVI, VI, UI, API
[5]	California, United States	Predict AQI for California	SVR with RBF kernel	Accuracy	(Data: API) CO, SO_2, NO_2, ozone, $PM_{2.5}$, temperature, humidity, wind
[30]	China	Measuring air pollutants with lidar signals	SVM LR RF BPNN	RMSE	(Data: Lidar) SO_2 ON, SO_2 OFF, NO_2 ON, NO_2 OFF
[31]	Germany	Predicting annual CH4 emission in farms	SVM RF MLR	RMSE MAE	(Data: Sensors) Methane, cattle, temperature, humidity
[32]	USA	Predicting NO_2 cattle emissions	Genetic algorithm	RMSPE	(Data: USDA EMET Lab) Lactating and dry cows, steers
[33]	Germany	Predicting NH_3 cattle emissions	GBM RF LRM SVM RMLR	RMSE MAE R^2	(Data: farm sensors) Cow count and mass, milk yield, temperature, humidity, CO_2, NH_3
[34]	Malaysia	Mapping air pollution	RF SVR	RMSE MSE	(Data: Himawari-8, Sentinel 5p) CO, HCHO, NO_2, O_3, SO_2, CH_4
[35]	United Kingdom	Retrieving AOT	ERDAS	MICROTOPS II photometer	(Data: Landsat) ozone transmittance, water vapor transmittance, aerosol scattering, surface reflectance
[36]	Germany	Mapping and identifying air quality patterns	GBRT	R^2; RMSE	(Data: MAIAC, MODIS, EEA) NOx, PM_{10}, $PM_{2.5}$, RH, SO_2, NH_3, temperature, moisture, images
[37]	China	Forecasting AOT	LSTM CNN	RMSE MAE	(Data: MODIS, MAIAC, MISR, OLI) CO_3, $PM_{2.5}$, SO_2, PM_{10}, O_3
[38]	India	Air pollution forecasting and AQI classification	Smotednn XGBoost RF SVM KNN	Accuracy RBF FPR FNR	(Data: NAMP) NOx, NO, SO_2, PM_{10}, $PM_{2.5}$, CO, O_3, NH_3, B, X, Touluene

Abbreviations: LR (logistic regression), LRM (linear regression model), ERDAS (Euclidean distance algorithm in ERDAS Imagine®).

Table 3. Literature survey of object detection methods for counting cattle.

Reference	Region	Purpose	Model	Metrics	Input Parameters
[39]	Asia, Europe, America, Africa	Counting in satellite images	FusionNet DRC SS-Net	MAE R^2	(Data: Google Earth API) RGB satellite images
[40]	South America	Counting and identifying livestock	CNN KNN RF	RMSE	(Data: Farm image sensors) RGB images
[41]	Brazil	Counting cattle	CNN	Precision, Recall F-measure	(Data: UAV) Drone images
[42]	Amazon	Tracking illegal cattle ranching	CSRNet LCFCN VGG16 FCN8	MAPE MAE	(Data: Maxar satellite) RGB images
Our study	United States	Cattle and air pollution correlation, counting cattle, AQI classification and mapping	Detectron2 YOLOv4 YOLOv5 RetinaNet LSTM CNN-LSTM Bi-LSTM	MSE RMSE MAPE MAE GAMPE MSLR	(Data: Google Earth API, Openweather API) RGB satellite images, CO, NH_3, NO, NO_2, O_3, PM_{10}, $PM_{2.5}$, SO_2

Abbreviations: VGG16 (VGG 16-layer net), FCN8 (FCN with 8 pixel stride).

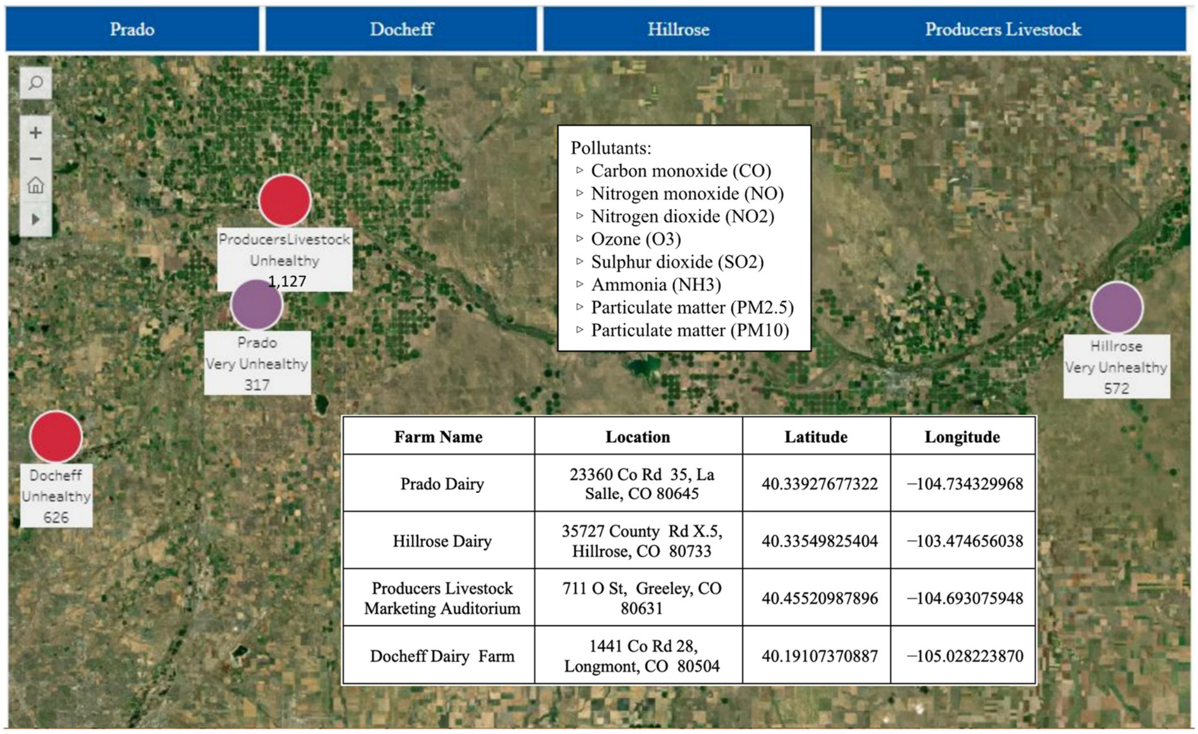
The literature survey on various machine learning and deep learning models for a variety of dataset types helped with understanding how to use the proposed models for object detection in stage 1 and air pollutant concentration prediction in stage 2 for our cattle farm air quality analysis project. For comparison with other studies, this study is included as the last row in Table 3.

3. Data Engineering

3.1. Study Area

For this study, we analyzed and evaluated air quality and environmental implications at selected animal farms using big data, deep learning, and machine learning models. The location chosen for this project is in Colorado, United States. To count the cattle, four farms in Colorado composed of beef and dairy farms were chosen: Prado Dairy, Hillrose Dairy, Producers Livestock, and Docheff Dairy Farm.

We chose to investigate the eight most common pollutants that were concentrated in cattle farms. In our investigations of the targeted farms, we found very few sources of emissions outside of livestock, the most notable being one to two tractors and cattle feeders on the farm. As the emissions values from these sources are very low and constant [43], we assumed that the farm emissions are primarily from livestock. Images from the satellite and pollutant and weather data from the API were collected for the four farms. The time range for this study was November 2020 to 15 August 2022. Figure 1 illustrates the AQI map visualization and our targeted pollutants and cattle farms. This study will educate farmers and anyone interested in the subject and increase public awareness of livestock greenhouse gas emissions.



Farm Name	Location	Latitude	Longitude
Prado Dairy	23360 Co Rd 35, La Salle, CO 80645	40.33927677322	−104.734329968
Hillrose Dairy	35727 County Rd X.5, Hillrose, CO 80733	40.33549825404	−103.474656038
Producers Livestock Marketing Auditorium	711 O St, Greeley, CO 80631	40.45520987896	−104.693075948
Docheff Dairy Farm	1441 Co Rd 28, Longmont, CO 80504	40.19107370887	−105.028223870

Figure 1. Map visualization of AQI of each pollutant for the cattle farms selected. Pollutants analyzed and cattle farm coordinates are included on the map.

3.2. Data Process

Data processing was considered as a significant step in this project as there were various types and sets of data used in order to achieve the goal of the project. The data required for the project were categorized into the following:

1. Cattle detection and counting, which requires satellite data;
2. Air pollutants based on coordinates, which requires remote sensing API.

The stages of the data processing included collecting raw data from a reliable source and splitting the data for training, testing, and validation purposes. The data processing methods for the categories mentioned above are explained below.

3.2.1. Satellite Data

In selecting the right data source to fetch the satellite images, we chose the Google Earth engine [44], as it has the numbers of satellites available and provides the time period of the data required. From the list of satellites, it was important to select a satellite with high resolution for cattle counting. After some research, we chose Sentinel-2 [45] and Landsat-8 [46] as they provide a greater resolution (30 m to 10 m resolution) and are the nearest imagery satellites for object detection. Figure 2 includes examples of satellite images taken from Google Earth engine.

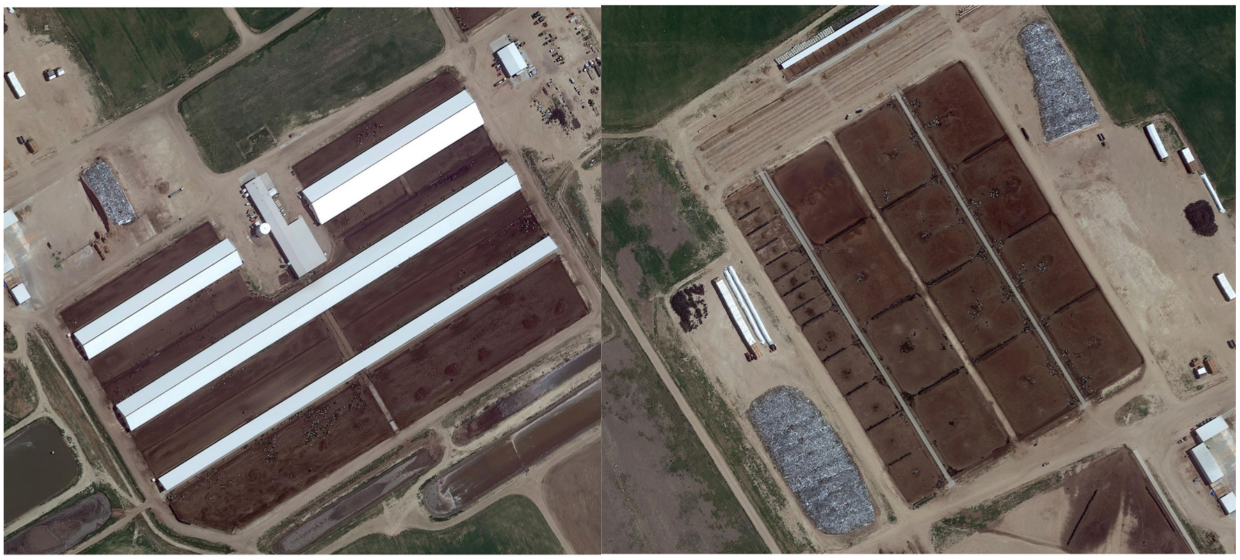


Figure 2. Example of satellite images before pre-processing.

3.2.2. Remote Sensing Data

To predict the air quality in cattle farms, it is mandatory to capture the pollutants level at each farm location. The coordinates of the four chosen farms were taken and pollutants levels were measured using remote sensing API. We used the free source OpenWeather API [47] to obtain the detailed level of each pollutant used in the project. The air quality provided by this API is in hourly granularity available from November 2020. As these are API data, not much storage was required from our end for the data processing. The coordinates are given to this API for testing purposes.

3.3. Data Collection

The data collection process for the four farms started with the cattle count in each farm, followed by collecting pollutant data and then weather data. The collection methods of each task are explained in detail below.

3.3.1. Cattle Count

The images from 15 March 2021 to 15 August 2022 were chosen from the Google Earth engine, and were split into training, validation, and testing data. The satellite images had a spatial resolution of 10 to 30 m and a zoom level of 100 m. A snippet was given to obtain the required data from the historical dataset. The farm coordinates and required time period were used to fetch high-resolution image data, which was provided by the Google Earth engine in collaboration with NAIP: the National Agriculture Imagery Program [48]. Once these details were given, the next step was to execute the program in Python to collect the data.

3.3.2. Air Pollutants

For this project, we collected the following parameters of the air pollutants from the OpenWeather API:

- Carbon monoxide (CO);
- Nitrogen dioxide (NO_2);
- Ozone (O_3);
- Sulfur dioxide (SO_2);
- Ammonia (NH_3);
- Particulate matter ($PM_{2.5}$);
- Particulate matter (PM_{10}).

From the above list of pollutants, nitrogen and Ammonia (NH_3) were the main metric, as cattle are the major contributors of these pollutants [49]. Table 4 shows the air quality index levels [50]. These values were later classified into different colors based on their health concern level, ranging from good to hazardous.

Table 4. Air quality index and classification.

AQI		**Pollutant Concentration**						
Category	**Value**	**NO_2**	**PM_{10}**	**O_3**	**$PM_{2.5}$**	**NH_3**	**CO**	**SO_2**
Good	0–50	0–53 (1 h)	0–54 (24 h)	0–54 (8 h)	0–12 (24 h)	0–200 (24 h)	0–4.4 (8 h)	0–35 (1 h)
Moderate	51–100	54–10 (1 h)	55–154 (24 h)	55–70 (8 h)	12.1–35.4 (24 h)	201–400 (24 h)	4.5–8.4 (8 h)	36–75 (1 h)
Unhealthy for Sensitive Groups	101–150	101–360 (1 h)	155–254 (24 h)	71–85 (8 h)	35.5–55.4 (24 h)	401–800 (24 h)	9.5–12.4 (8 h)	76–185 (1 h)
Unhealthy	151–200	361–649 (1 h)	255–354 (24 h)	86–105 (8 h)	55.5–150.4 (24 h)	801–1200 (24 h)	12.5–15.4 (8 h)	186–304 (1 h)
Very Unhealthy	201–300	650–1249 (1 h)	355–424 (24 h)	106–200 (8 h)	150.5–250.4 (24 h)	1201–1800 (24 h)	15.5–30.4 (8 h)	305–604 (24 h)
Hazardous	301–500	1249–2049 (1 h)	425–604 (24 h)	405–604 (1 h)	250.5–500.4 (24 h)	1800+ (24 h)	30.5–50.4 (8 h)	605–1004 (24 h)

In order to collect the pollutant data from the API, an {API key} was first generated from the OpenWeather website. Once the {API key} was received, the API key and location coordinates (latitude and longitude) were used for collecting the pollutants level.

3.4. Data Pre-Processing

The next step was to remove any anomalies in the data and prepare the data for smooth model training. As this process involves different types of datasets, preprocessing was required for all of them. After finding geographical data for each of the four chosen farms, we used API to extract pollutant data for each farm. This was performed in Google Colab. The raw dataset contains columns with the hourly values of each pollutant, longitude and latitude, cattle count, and date time. The format of date time was changed from timestamp to date and hour. We merged all four extracted datasets into one and labeled images for training the model for cattle detection and counting with Roboflow [51] for image annotation.

Figures 3 and 4 are examples of satellite images after they were annotated in Roboflow. After annotating the images, we auto oriented them and transformed the cattle farm images into tiles for better model understanding. The augmentations used to generalize the object detection models were horizontal and vertical flipping, shearing with ±15 degree horizontal and vertical, brightness between −25% to +25%, blur and noise up to 2.5 px and 3% of pixel respectively, bounding box rotation between −15 and +15 degrees, bounding box shear ±15 degree horizontal and vertical and mosaic.

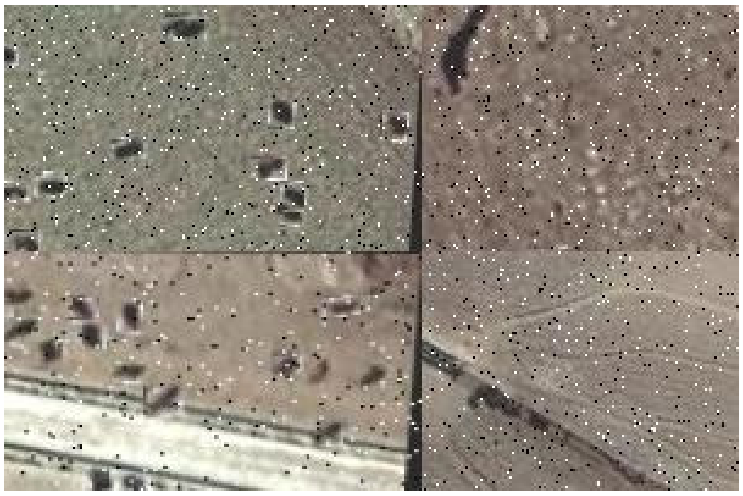


Figure 3. Annotated version of a satellite image taken during the day.

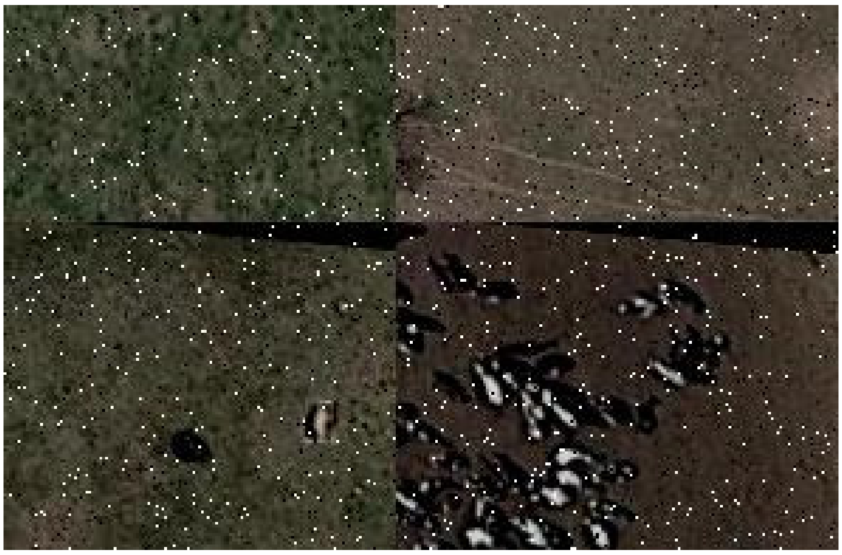


Figure 4. Annotated version of a satellite image taken during the night.

3.5. Data Transformation

The remote sensing and satellite image data were transformed to make it organized so that it would be quick and easy for the different models in each stage of modeling to work on. During data transformation, pre-processed data were transformed into a particular dataset. Improving the quality and variety of the training dataset is critical for the performance and ability of the model to generalize. The data transformation steps differed for satellite image data for object detection in stage 1 and remote sensing data for air pollutant concentration prediction in stage 2.

3.5.1. Stage 1—Satellite Imagery Dataset

For stage 1, we collected satellite image data for each of the four different farms selected in Colorado to detect cattle and calculate their total count. For the transformation of the data collected, we increased the size and variety of the training dataset by including more transformed versions of existing data in the training dataset. By reproducing existing images in different orientations, we fed the model additional information to learn from without having to gather and label more training data. The sample image data for this project employed horizontal flipping, vertical flipping, random brightness augmentation, random cropping, and rotation by no more than 45 degrees to chosen images from the dataset. Since the cattle in the images were in different positions, our model will be able to better handle changes in object orientation when using rotation as an augmentation technique. Noise in satellite images was removed as part of data smoothing. A binning algorithm was implemented in this data transformation stage where the entire farm area was split into bins and smoothing of each of the bin regions was performed. Figure 5 shows the satellite image before binning, and Figure 6 shows the same image after binning.

3.5.2. Stage 2—Remote Sensing Data

For stage 2, remote sensing data were collected from OpenWeather API for the four different farms selected to predict the next hour air pollutant concentration in cattle farms for the eight pollutants selected. For each timestamp of hourly air quality, the remote sensing dataset was combined into different time periods such as a week, month, season and year. New features that were added to the dataset, such as the latitude and longitude, were encoded for further usage. As we needed data in ascending sequence of time, the dataset needed to be sorted in the datetime index. Since diverse scales of variables do not contribute equally to model fitting and learning, resulting in a bias, it was important to normalize the dataset. StandardScaler was used for normalizing the dataset. Due to the fact that different scales of variables contributed differently to fitting to the model and learning, resulting in bias, it was critical to normalize the dataset. We used StandarScaler() in the sklearn.preprocessing package to normalize our stage 2 dataset. After the values dataset had been normalized and standardized, the value of all the images in the array was between zero and one. As part of data standardization, the image features were converted to have zero mean and unit variance as seen in Figure 7. This conversion helped the model have no bias during the gradient computation and also helped the model coverage be better.

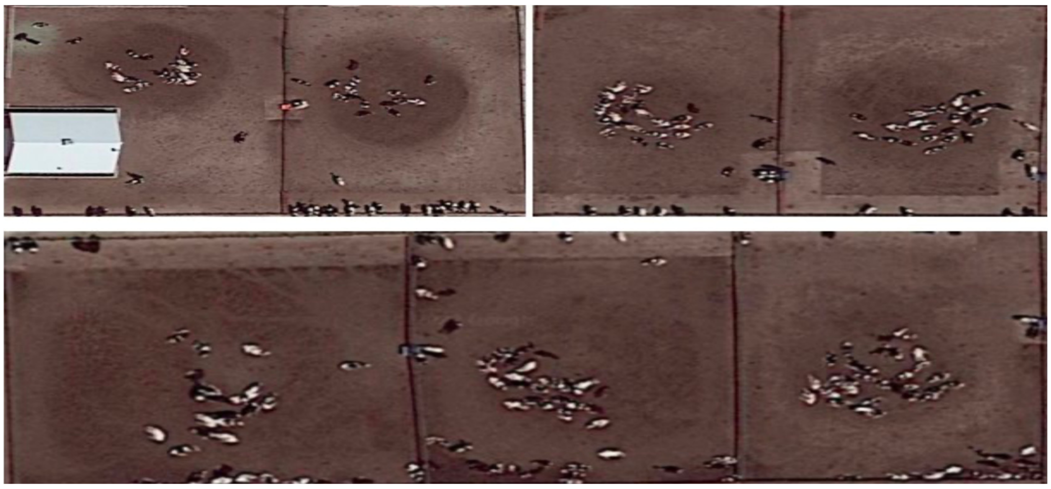


Figure 5. Sample satellite image data before binning.

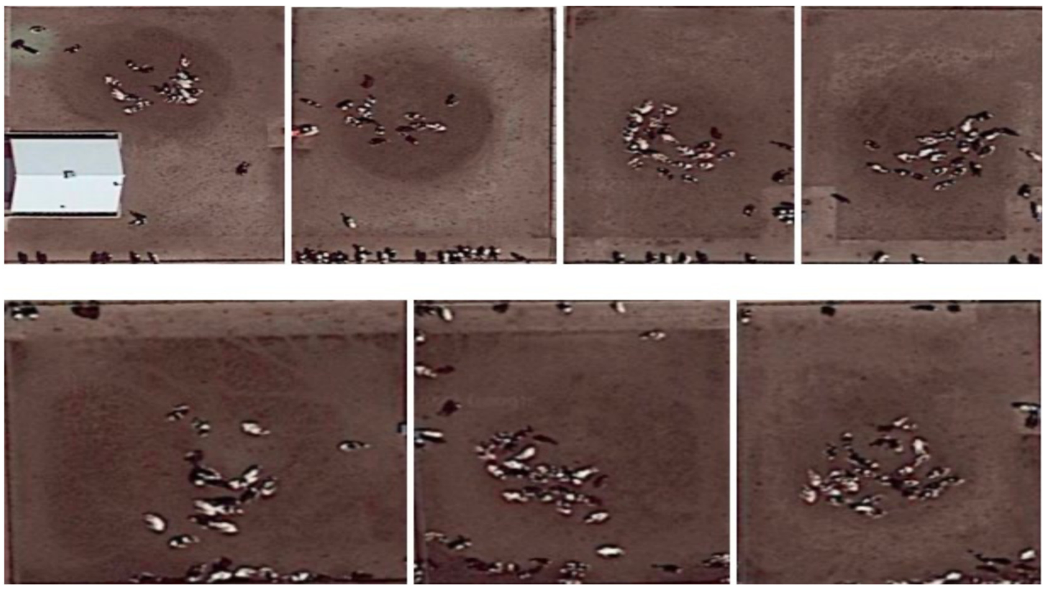
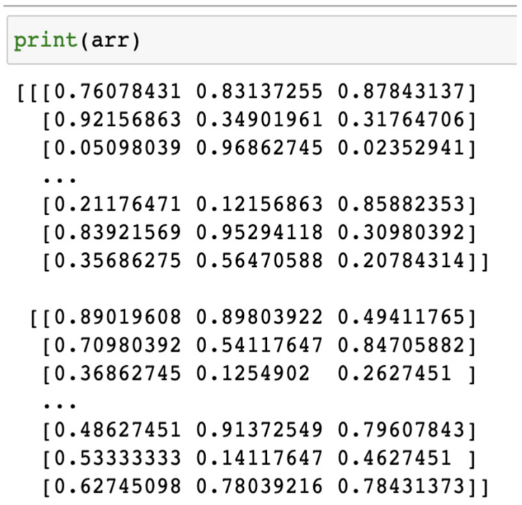


Figure 6. Sample satellite image data after binning.

3.6. Data Preparation

To prepare the data, we generated training, validation, and testing datasets to avoid overfitting and to obtain a proper model accuracy. This step was crucial for the further training of our models as it is an important factor while training the model. The number of features involved in training the model was directly proportional to the complexity of the model. Not all features were needed during the training and testing process, and those such as longitude and latitude could be removed. It is necessary to have a system in place to determine how effectively the model generalizes. The input image dataset was split into training and testing data. Training data contained the training dataset and the validation dataset. Data were divided into 60% training, 20% validation, and 20% testing data using

Roboflow. Validation parameters were evaluated after each training epoch to monitor the performance of the model and see whether it had reached optimum performance with the validation set. Training, validation, and testing datasets used preprocessed images. Only test and validation datasets used ground truth and images with augmentation were only in the training dataset.



```
print(arr)
```

```
[[[0.76078431 0.83137255 0.87843137]
  [0.92156863 0.34901961 0.31764706]
  [0.05098039 0.96862745 0.02352941]
  ...
  [0.21176471 0.12156863 0.85882353]
  [0.83921569 0.95294118 0.30980392]
  [0.35686275 0.56470588 0.20784314]]

 [[0.89019608 0.89803922 0.49411765]
  [0.70980392 0.54117647 0.84705882]
  [0.36862745 0.1254902  0.2627451 ]
  ...
  [0.48627451 0.91372549 0.79607843]
  [0.53333333 0.14117647 0.4627451 ]
  [0.62745098 0.78039216 0.78431373]]
```

Figure 7. Remote sensing data after conversion to zero mean and unit variance.

Because weather and air quality datasets are time series, they could not be randomly divided and were instead divided by timestamp and trained using time-based splitting. The time-series data utilized the unix time system for each image since other time formats would lead to an unreliable result. Weather and remote sensing datasets contained data for the time interval. Divided datasets needed to be reshaped for each used model to meet its requirements.

The images were annotated with Roboflow before feeding them to the models. Roboflow annotated the images quickly while keeping track of the annotated and unannotated images, and split the images into training, testing, and validation sets as illustrated in Figure 8. This was all carried out in Google Colab with Roboflow API, which helped to maintain the code neatly. The images were then flipped and rotated with other augmentation techniques.

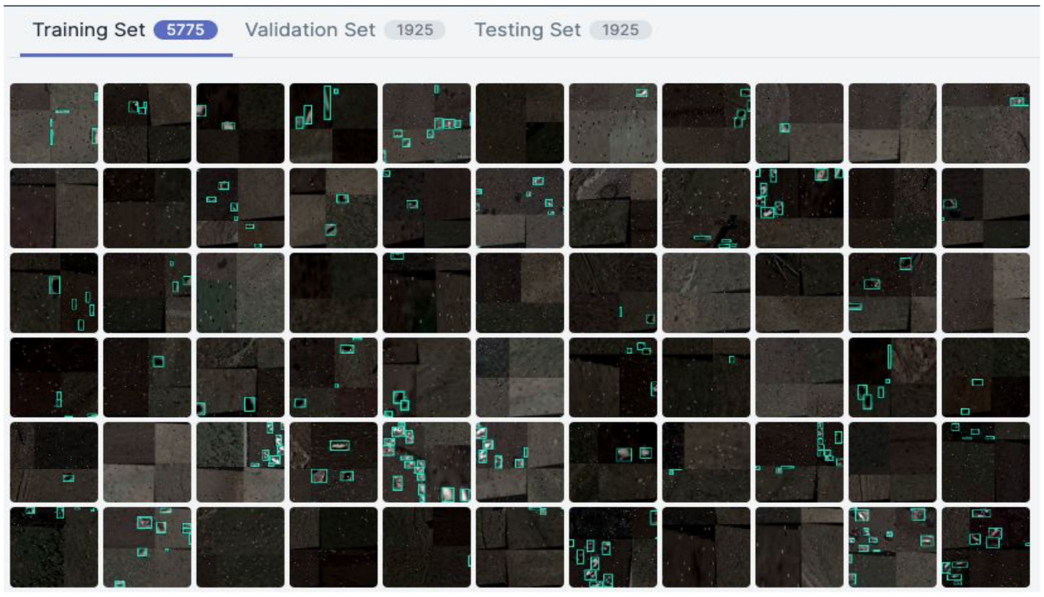


Figure 8. Annotated data in Roboflow split into training, validation, and testing sets.

3.7. Data Statistics

The satellite image dataset, weather data, and hazardous gases data for the four chosen farms were collected and saved in the required format files. The remote sensing data containing air pollutants were collected, pre-processed, transformed, and prepared. Different image rotations and cattle densities may be observed in the given images and their flipped counterparts. Figure 9 shows a spider map with various pollutants and their concentration across months for the farms. These spider maps were compared for several pollutants for the months of April, July, October, and February.

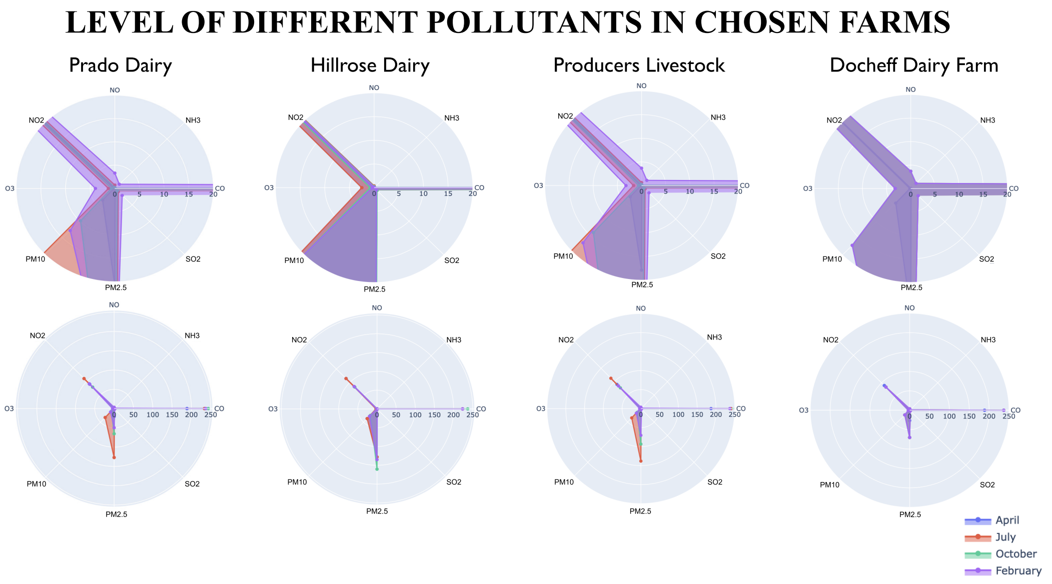


Figure 9. Spider map for levels of different pollutants in chosen farms.

Figure 10 shows the correlation between variables—the brighter the color, the higher the correlation between variables. We can see a high correlation between $PM_{2.5}$ and PM_{10}. We can see the correlation between each of the pollutant concentrations with respect to the cattle count.

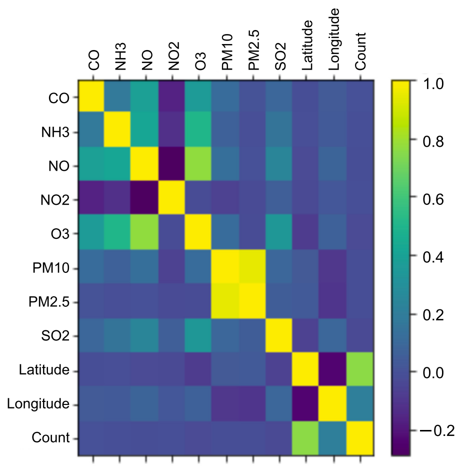


Figure 10. Covariance matrix.

Figure 11 shows the density distribution of each of the pollutants. There is a dense concentration for NO_2 from 50 to 150, indicating medium concentrations. However, for most of the other pollutants, such as CO, NH_3, NO, O_3, PM_{10}, SO_2, $PM_{2.5}$, there is a spike at the beginning, indicating smaller concentrations for those respective pollutants. This allows us to get an idea of the values for each of the pollutants for further analysis.

The below Figure 12 shows the trend of pollutant concentrations over a year. All these pollutants are displayed with different colors for better comparison. As we can see, there is a spike in the first quarter of CO, NH_3, and NO, and the opposite can be observed for PM_{10} and $PM_{2.5}$, where the spike is towards the end. Then, it settles to a lower side and it is much more random throughout the year for NO_2. High SO_2 is present from the middle of the term.

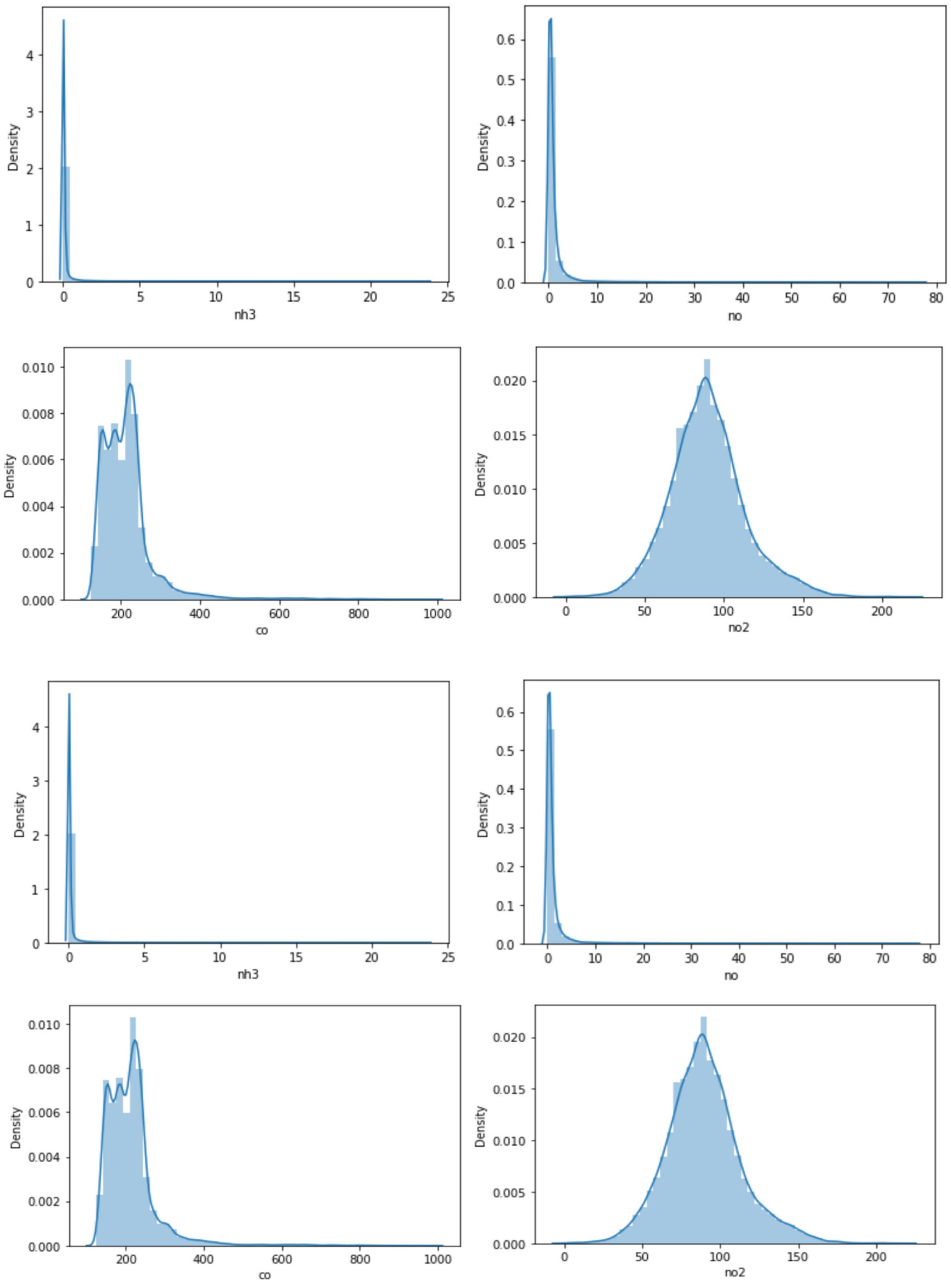


Figure 11. Density histogram.

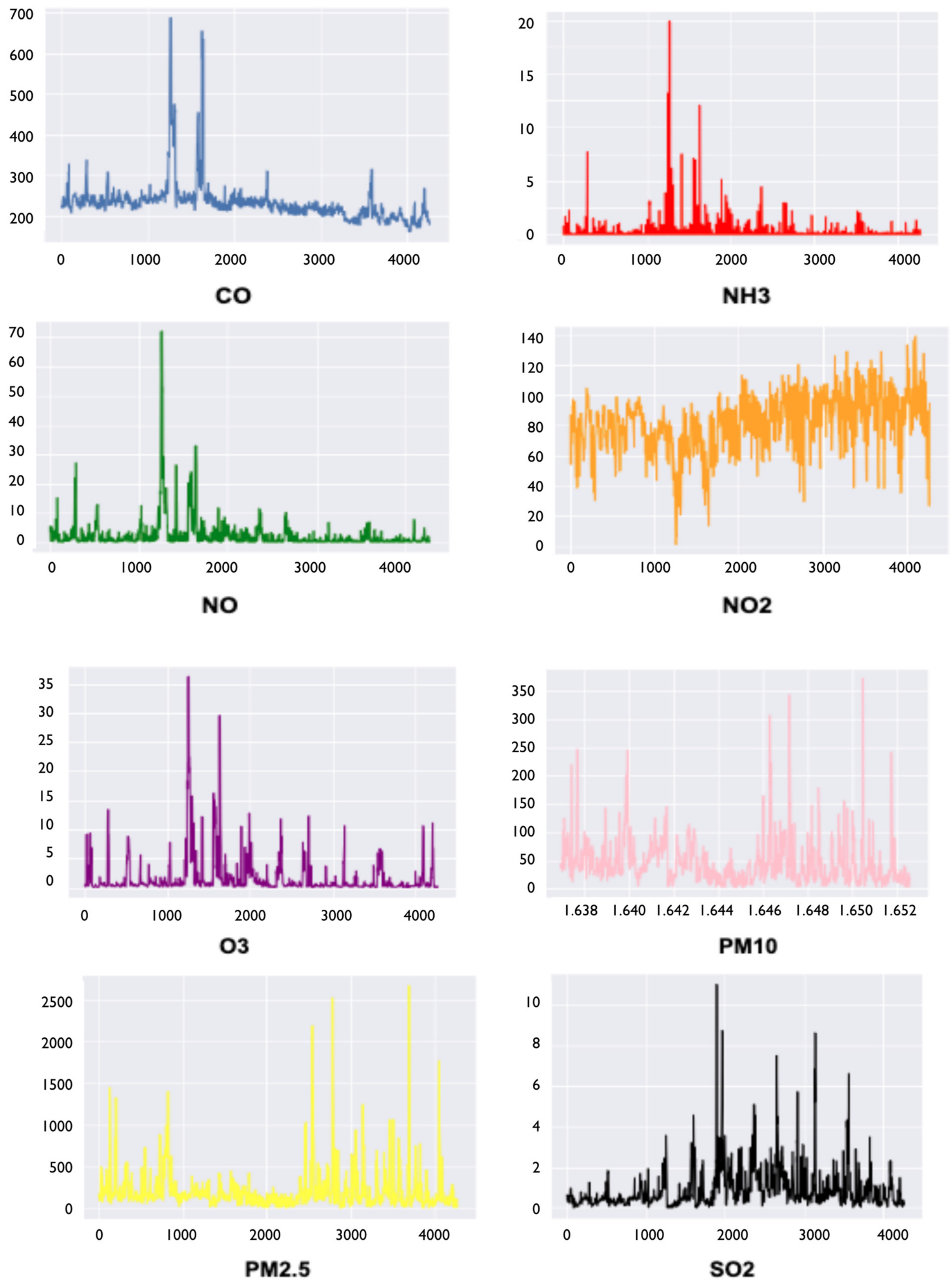


Figure 12. Individual pollutant concentration trend across a year.

4. Model Development

4.1. Model Proposal

The proposed cattle farm air quality analyzer has three facets, which are counting cattle, predicting pollutants, and visualization based on the pollutant severity. A satellite dataset collected from the Google Earth engine was used to train the stage 1 models for counting the number of cattle in a satellite image. Using weather and remote sensing data, a stage 2 model was developed to predict the presence of different pollutant concentrations in particular cattle farms. Once the prediction was made, these data were visualized on a map with the help of Tableau [52] to show the severity of each of the pollutants. We proposed multiple machine learning/deep learning multivariate models for time series data to provide better evaluation results in this project.

In the first stage model, the most important part for counting the cattle is the object detection. For object detection, this project used four models—YOLO v4 [53], YOLO v5 [54], ResNet [55], and RetinaNet [56]. For YOLOv4, the input of the initial convolutional layer dimension was 416 × 416 × 3 on which 32 filters were applied. These are the standard dimensions for YOLOv4 and create a balanced model. For YOLOv5, the input dimensions for the first convolutional layer started with 64 × 64 × 3 with initial 3520 parameters. This model was built with eight convolutional layers. The next model used for object detection and counting was ResNet, which was pre-trained using ImageNet [57]. We modified ResNet50 with two FC layers, which were connected with 512 neurons. These modifications were made to better count the cattle and reduce the vanishing gradient problem. Furthermore, the Detectron2 model developed by Meta was used as a part of ResNet for cattle counting [58,59]. Our Detectron2 model had an input dimension of 256 × 256 × 3 with deep convolutional layers. RetinaNet, a bottom-up pathway and one-stage detection model that uses a focus loss function, was used for calculating the cattle count [10]. The input dimensions on the first convolutional layer were given as 64 × 64 × 3.

Once the stage 1 models had been completed, stage 2 models were used for the pollutant prediction. We implemented a combination of RNN models, including CNN-LSTM, Bi-LSTM, and the multiple lag timestep LSTM. The CNN-LSTM model was used for pollution prediction based on the last 24 h of pollutant data, and predicted the next hour data with training using remote sensing data. The modeling was performed by creating 64 filters on the convolutional layer with the activation function ReLU, and was then connected with an LSTM layer followed by a fully connected layer to produce the prediction results. The batch size for this model was kept as 64 with 2176 parameters and ran for 12 epochs. For the Bi-LSTM model, two layers of LSTM were added—a forward LSTM layer and a backward LSTM layer. The output from the Bi-LSTM was concatenated, after which the prediction results were provided by the model. The batch size of the Bi-LSTM model was kept as eight with a window length of 64 and was deployed for 50 epochs.

In order to predict the pollutant concentration for each of the specified farms, support vector regression (SVR) [60], linear regression, and convolutional neural networks (CNN) [61] were proposed. For SVR, the maximum error parameter was defined and fine-tuned to increase the accuracy of our model. The linear regression model attempts to identify potential associations between two variables by fitting linear equations to the dataset, and calculate best fit by minimizing the squared errors [50]. For CNN, the input data were resized and fed to a convolution layer where the ReLU activation function was followed for all the layers except the last layer, which used sigmoid activation. CNNs are best with image data for analysis, but these can be extended for predicting by adding regression layers at the end of the network after a convolutional network for regression tasks [62].

Figure 13 outlines the proposed model architecture flow for this project, showing the first stage modeling for counting cattle with the proposed YOLOv4, YOLOv5, Detectron2, and RetinaNet models, and the second stage of modeling with models CNN-LSTM, Decision tree regressor, support vector regressor, and Bi-LSTM for air quality prediction.

Models CNN-LSTM and Multilag LSTM were used as stacked models to provide high accuracy. The output of the first stage was fed as input to the stage 2 models along with the remote sensing data to predict the pollutant concentration. The output from all the stage 2 models was used to classify the pollutants predicted based on their severity for visualization, and this output was displayed using Tableau, which was then fetched to the web application. Figure 14 shows the pipeline of the system.

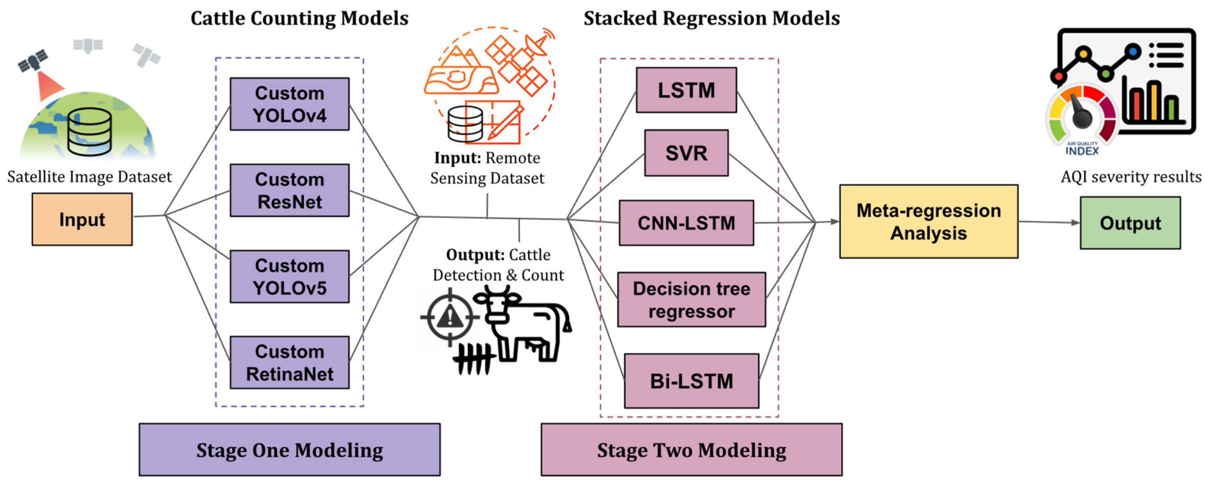


Figure 13. Modeling architecture flow for proposed models.

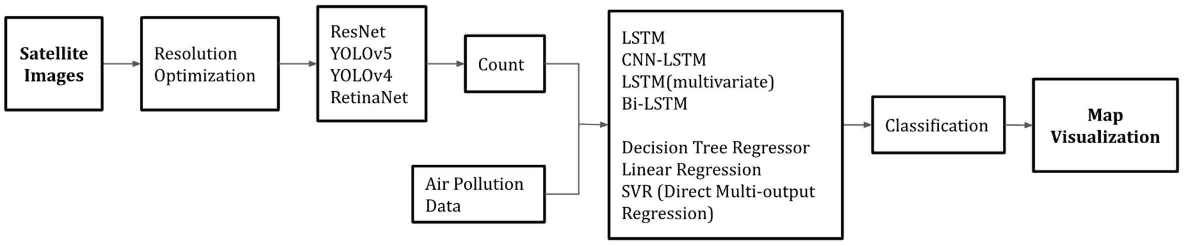


Figure 14. Pipeline of the system.

4.2. Model Supports

In this project, there were a large amount of data that were hard to store and process on a local machine. Thus, Amazon S3 was used as support to store the dataset for all the machine learning and deep learning approaches proposed, including SVR, linear regression, random forest, linear regression, Bi-LSTM and CNN-LSTM. Amazon Web Service (AWS) was also used for the storage of data, which were retrieved and stored using the secured Hypertext Protocol Transfer Secure (https). All the data collected online were migrated to the S3 bucket and were extracted from the stored database when preparing for modeling. Detectron2, RetinaNet, YOLOv5, LSTM, Bi-LSTM, and CNN-LSTN were run with GPU Nvidia K80/T4 and memory of 12 GB/16 GB. YOLOv4 was run with GPU: Tesla T4 with memory of 12 GB. Linear regression, DTR, and SVR were run with CPU: Intel(R) Xeon(R) with memory of 12 GB. All models were run in Google Colab, written in Python 3, and developed using keras, Tensorflow, Pytorch, and scikit-learn packages.

For stage 1 modeling, data preprocessing was performed using Roboflow. Augmentation techniques such as flip, shear, brightness, blur, mosaic, noise, bounding box: shear, and bounding box: rotation were supported by Roboflow for this project. For the second modeling stage, the hourly air pollutant data were preprocessed with the help of the Google Colab platform in Google Drive using Python3. These platforms were considered because they provide shared and up-to-date access across all the team members. A selection technique used for image segmentation was performed by comparing the evaluation metric Intersection over Union (IoU) score of the different image segmentation models, YOLOv4, YOLOv5, ResNet and RetinaNet. All models were trained in Python 3.7, since it has the built-in libraries we required for a variety of computational and modeling problems. Pytorch was used for easier data loading and processing for all the ML and deep learning models proposed. For the first stage, we trained different models and picked the most accurate one, which took periodic satellite image data of cattle farms as input and helped to count the number of cattle in each farm. After preprocessing of the input, data were split into train, validation, and test sets in the ratio of 60:20:20 for model training and evaluation. New data were given after training for the model to count the cattle in a new image accurately. Figure 15 illustrates the architecture for the first stage of modeling.

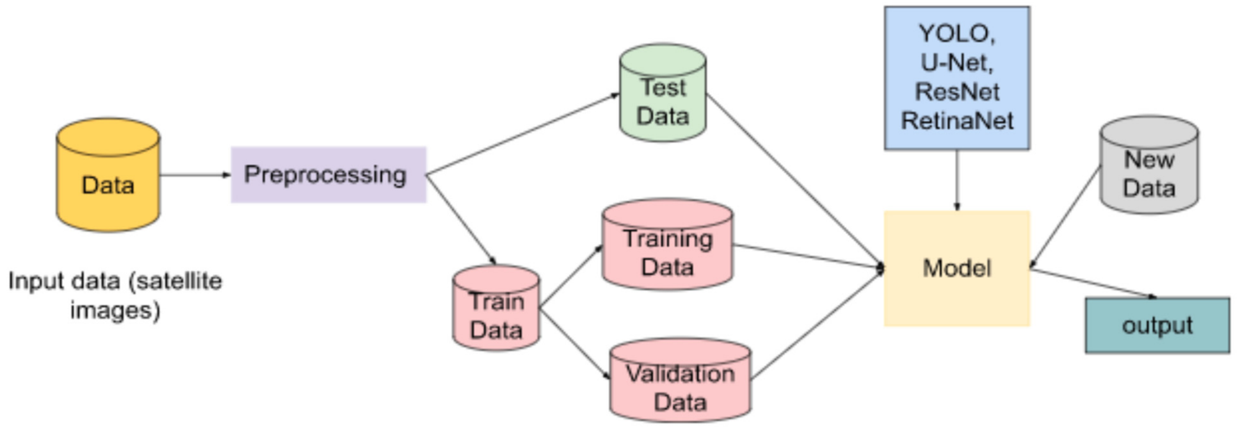


Figure 15. DL data flow for stage one modeling.

For the second stage of modeling, data from other sources, such as the farm data, weather data, and pollutant data, formed the input to the model. Similar to stage one, we split the data for training, validation, and testing. In the stage two model, we used the previous model outputs, such as cattle count, as one of its input features to best predict the pollutant concentration and classify them based on their severity. The proposed models—support vector regression (SVR), linear regression (LR), LSTM (RNN), CNN-LSTM, Bi-LSTM, and decision tree regressor—were used in the prediction of air pollutants in this stage of modeling shown in the ML data flow diagram below. Figure 16 illustrates the architecture for the second stage of modeling.

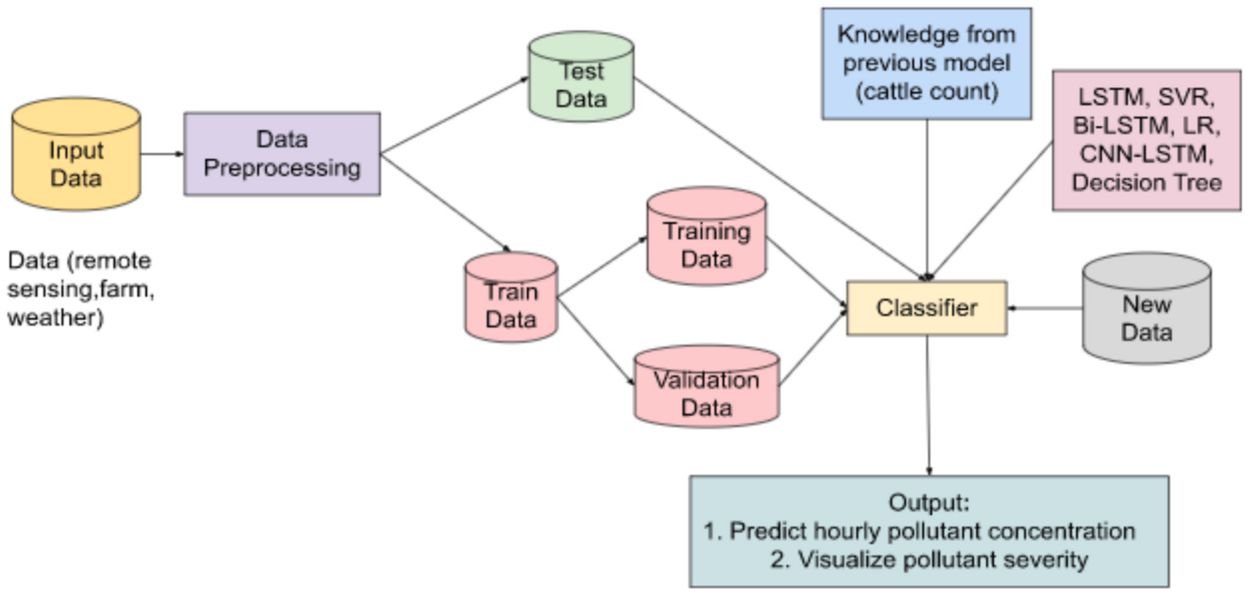

Input Data
Data (remote sensing,farm, weather)
Data Preprocessing
Test Data
Train Data
Training Data
Validation Data
Knowledge from previous model (cattle count)
LSTM, SVR, Bi-LSTM, LR, CNN-LSTM, Decision Tree
Classifier
New Data
Output:
1. Predict hourly pollutant concentration
2. Visualize pollutant severity

Figure 16. ML data flow for stage two modeling.

4.3. Model Comparison

We compared various models for both stages including Cubist [63], eXtreme Gradient Boosting algorithms [64], CSRNet [65], logistic regression [66], YOLOv5 [67], and Fuzzy KNN [68]. For comparing models that have used counting methods using images retrieved from UAV, we considered various algorithms including ResNet, YOLOv4, Mobilenet—SSD, RetinaNet, and YOLOv5. Detecting livestock is a crucial stage in the cattle counting process and, when trying to anticipate the concentration of gasses across the farm, the projection of the number of cattle provided us with an important factor to take into consideration. The improved model YOLOv5 has a similar structure to YOLOv4. In detection problems, the common methodology is that all inputs are of the same size and dimension, but in YOLOv5, the borders are ignored and the focus is more towards the detection part which in turn increases target detection speed. YOLOv5 added a new focus mechanism to our model and was the backbone for target detection. CSPnet enhances the ability of feature integration and, instead of using CIOU_Loss as with YOLOv4, YOLOv5 uses GIOU_loss. RetinaNet utilizes Focal Loss, which neutralizes the problem of class imbalance. The feature pyramid in RetinaNet helps detect objects at multiple scales [69]. For our purpose, the YOLOv5 mechanism gave the best converging rate due to its method of adding positive sample anchors to boost convergence and training efficiency, and is significantly smaller than YOLOv4 [70]. The model has three main modules: a backbone for feature extraction from images; a neck for combining extracted features from the previous module and passing to the next module; and a head module that takes features from the neck network and makes predictions. YOLOv5 also has a better inference time than YOLOv4, and results in higher average precision and recall than the other models we tested. After comparing the different models, we decided to implement YOLOv4, YOLOv5, ResNet, and RetinaNet for stage one modeling. Table 5 provides an overview and comparison of each model we used in stage one and stage two.

Table 5. Machine learning model comparison—Stages 1 and 2.

Model	Mechanism	Advantage	Disadvantage
YOLOv4	It splits the input image in m-sized grids and for every grid it generates 2 bounding boxes and classes with probabilities	It is fast and open source. Classifying images in real time is faster and more accurate than other algorithms.	Spatial constraints are strong, two grid cells only predict a single class at a time.
YOLOv5	Uses auto anchor boxes, mosaic augmentation, scaling, adjust colors, combines sliced images into one and finds new classes.	Provides better converging rate, is faster, smaller, implements new findings, shows good performance in real- time detection, and gets higher accuracy.	Has limited literary support, predicts single class at a time.
ResNet	Multiple layers of plain networks with a shortcut connection that creates a residual network.	Deeper training of the network, minimizing the information loss issue. Identity mapping for vanishing gradients.	The model training process is time-consuming.
RetinaNet	Is a unified network consisting of a main network and two specialized networks for different tasks,	Takes on the difficulty of detecting small and dense things. Solves the class-imbalanced problem. Is fast and accurate.	More suitable for when a greater mean average precision in recognition is needed.
LSTM Single step	Stores or writes information by using a gating mechanism to read.	Learning long-term dependencies, in backpropagation, solves the problem of vanishing gradient.	It requires a long training time, is easy to be overfitted, and takes a lot of memory.
LSTM Multi step	Similar to LSTM, it considers multiple influenced factors.	Predicts several outputs simultaneously, is suitable for short-period predictions.	Is under the presumption that the time series is conditionally Gaussian.
CNN-LSTM	Selectively remembers patterns for a longer period of time where CNN is used to extract time features.	Provides a wide range of parameters (learning rate, input & output bias) for tuning. Handles vanishing gradient problem.	Time duration for training to solve real world problems needs time. Also prone to overfitting and requires memory to be trained.
Bi-LSTM	Gets input from both sides and examines sequences front-to- back and back-to- front.	Provides a past and future context.	Is a costly model due to the additional LSTM layer, long training time, slow model.

For building a cattle farm air quality analyzer, various models were compared to find the best fit for the collected dataset from various sources. Model comparison was carried out in different categories depending on the use case, namely: regression models, classification models, models using satellite data, models used in counting methods, and models using remote sensing data. When comparing models that have used remote sensing data for predicting air pollution, we had to consider various features and labels for these pollutants. We compared Cubist and eXtreme Gradient Boosting algorithms from the paper [64], with Cubist performing the best. We also compared different models used in other papers such as CNN-LSTM, LSTM, LSTM (Multivariate), CNN, and BiLSTM. For cattle farm pollution prediction, we considered CSRNet, LCFCN, genetic algorithms, linear regression, and support vector regression models. These models predict the future trends to estimate the pollution across the farm. For classifying the predicted pollutants based on their severity, we used ML classification models such as Random Forest, logistic regression, K Nearest Neighbor, and Fuzzy KNN. The output was displayed on heat maps. After comparing the different models, we decided to implement LSTM single step and multi step, CNN-LSTM, and Bi-LSTM for stage two modeling.

4.4. Model Evaluation

A comparison of models was made with the help of a validation set, and we selected our model from the evaluation metrics results on testing data. For comparing models that use remote sensing data for predicting air pollution, the best evaluation metrics are usually MSE [71] and RMSE [72]. MSE is not robust to noisy data so we considered RMSE instead, where RMSE values were inversely proportional to model accuracy [65]. For stage 2 modeling, proposed regression models such as SVR, linear regression, and CNN use these metrics. For comparing models that have prediction models, such as CNN, linear regression, and support vector regression, and for the cattle farm pollution, comparing models such as CNN-LSTM that have used satellite data, we used a 10-fold cross validation in which we divided the data into subsets and then ran them on all models. The CV-RMSE and CV-R2 were evaluated to identify the models with the best accuracy [28]. CV-RMSE gives us a value in absolute terms while CVR2 tells us how well a model can predict in terms of percentage. For comparing models that utilize counting methods using images, we used mean absolute percentage error (MAPE) to differentiate between sparse regions and densely populated regions, and grid average mean absolute percentage error (GAMPE) to measure localization performance and counting the cattle.

For the classification models linear regression and Random Forest, we used a confusion matrix [73] to understand the performance of our model and to check its ability to predict either true or false for given data. For counting the number of cattle in a given image as part of the first stage of modeling, models such as RetinaNet and YOLO use the Intersection over Union (IoU) [74] evaluation metric for performance comparison of the image segmentation. Equation (1) shows the calculation for the IoU metric [75].

$$\text{IoU} = \text{TP}/\text{FP} + \text{TP} + \text{FN}. \tag{1}$$

The evaluation metrics that were calculated for model performance in the second stage of modeling included accuracy, sensitivity, specificity, and precision [76]. The formulas for each metric are given below [77].

$$\text{Accuracy} = \text{TP} + \text{TN}/\text{TP} + \text{TN} + \text{FN} + \text{FP};$$

$$\text{Sensitivity (Recall)} = \text{TP}/\text{TP} + \text{FN};$$

$$\text{Specificity} = \text{TN}/\text{TN} + \text{FP};$$

$$\text{Precision} = \text{TP}/\text{TP} + \text{FP}. \quad (2)$$

The F1 score shown in Equation (3) combines recall and precision in a single metric while giving equal weight to both, which in turn provides a greater interpretation of our model and balances the imbalanced data [78]. It is mainly used in classification problems as part of the second stage of modeling when classifying pollutants based on their severity [79].

$$\text{F1} = 2 * (\text{precision} * \text{recall})/(\text{precision} + \text{recall}). \quad (3)$$

The Receiver Operating Characteristic (ROC) curve is used to determine the performance of classification models at various threshold settings. For our classification purpose, it was essential to understand how the model performed while classifying cattle into different classes, as a wrong classification might result in erroneous results for predicting the emitted gasses from cattle. The Area Under the ROC Curve (AUC) [80] is used to give the average performance across thresholds and measure the quality of predictions and their ranking. The higher the AUC, the better the performance of the model. MAE (mean absolute error) is used to evaluate model accuracy, and was calculated separately for each pollutant. A lower MAE value corresponds to a better model and accuracy [81]. Equation (4) shows the formula for MAE.

$$\text{MAE} = 1/n * \Sigma \mid yi - \hat{y}i \mid . \quad (4)$$

4.5. Two-Stage Model Experimental Results

4.5.1. Stage One Modeling Results

For the first stage of modeling, we sourced the satellite image data from the Sentinel-2 satellite. Figure 17 shows the evaluation results from the four different object detection models—Detectron2, YOLOv4, RetinaNet, and YOLOv5. We can see the cattle being detected by each of the different models using bounding boxes. These bounding boxes have an object accuracy score for each object detected in the image.

Table 6 below shows the average precision and average recall metrics values for each of the proposed stage 1 deep learning models. It is observed that YOLOv5 gave the best results and Detectron2 performed poorly compared to the other models.

Table 6. Average precision and recall for stage 1 modeling.

Models	Average Precision	Average Recall
Detectron2	0.871	0.075
Yolov5	0.916	0.912
Yolov4	0.872	0.879
RetinaNet	0.881	0.887

Detectron2

For the Detectron2 model, the fine-tuning was accomplished with changing the batch size from 32 to 64 and increasing the iterations from 1500 to 3000. When running with this final version, a greater performance was witnessed with less training time. Initially, the training time was around 1 h and 53 min. But, the current time is around 2 h and 58 min as the iterations are doubled. The average precision improved from 0.702 to 0.871, and average recall from 0.710 to 0.875.

YOLOv4

In the case of YOLOv4, which was cloned from Darknet, we tuned the model by increasing the batch size from 32 to 64, adding dense layers, and increasing iterations from 500 to 2000. When running with this final-tuned model, a greater performance was witnessed with a slight increase in training time. Training time was initially around 1 h and

3 min with 500 iterations and the current time is around 1 h and 58 min with 2000 iterations. The average precision of 0.773 is now 0.872, and the average recall of 0.727 is now 0.879.

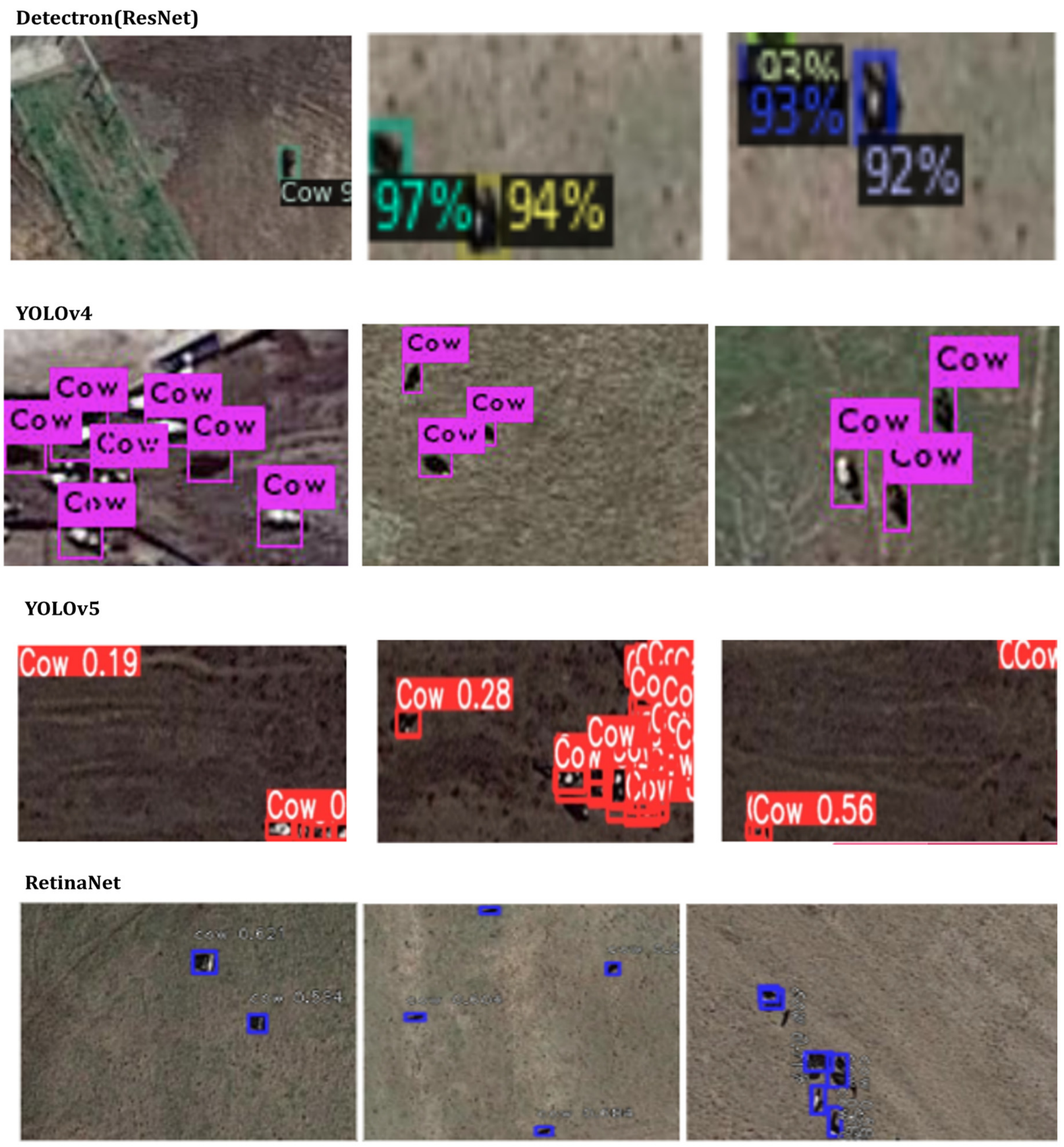


Figure 17. Cattle detection on satellite images for stage 1 modeling.

YOLOv5

YOLOv5 was initially trained with 150 epochs and the training time was noted to be approximately 1 h and 12 min. After fine tuning, the number of epochs was increased to 650. The final model took a training time of around 3 h and 59 min. Although the training time was extended, the model had a significant increase in average precision and recall, with average precision increasing from 0.782 to 0.916 and average recall increasing from 0.727 to 0.912.

RetinaNet

The RetinaNet model was initially trained with 80 epochs with a training time of 2 h and 29 min. During the fine tuning, the model was deployed with 300 epochs and training time was increased to 4 h and 18 min. This resulted in an improved average precision of 0.881 compared to the initial 0.776, and an average recall of 0.887 from 0.729.

4.5.2. Stage Two Modeling

The figures below show the prediction results for the LSTM model, where the blue curve indicates the training loss and the orange line represents the validation loss curve. The left graph in Figure 18 displays the train and validation errors for the single lag multivariate LSTM where the train error is seen to be more than the validation error. This shows that the model has learnt the input dataset well and is overfitting. The right graph on the figure illustrates the LSTM multiple lag timestep loss curve where both the train and validation errors are more or less similar. Figure 19 shows the graph of the CNN-LSTM model's performance, where we can see that there is a steady difference between the blue curve showing training loss and the orange validation loss curve, making it a good model to work with. The actual and predicted curves show several small spikes in the middle but are achieving similar values by the 15th epoch. Similarly, the right graph on the figure shows the training and testing loss curve for Bi-LSTM. All models used mean absolute error for loss calculation.

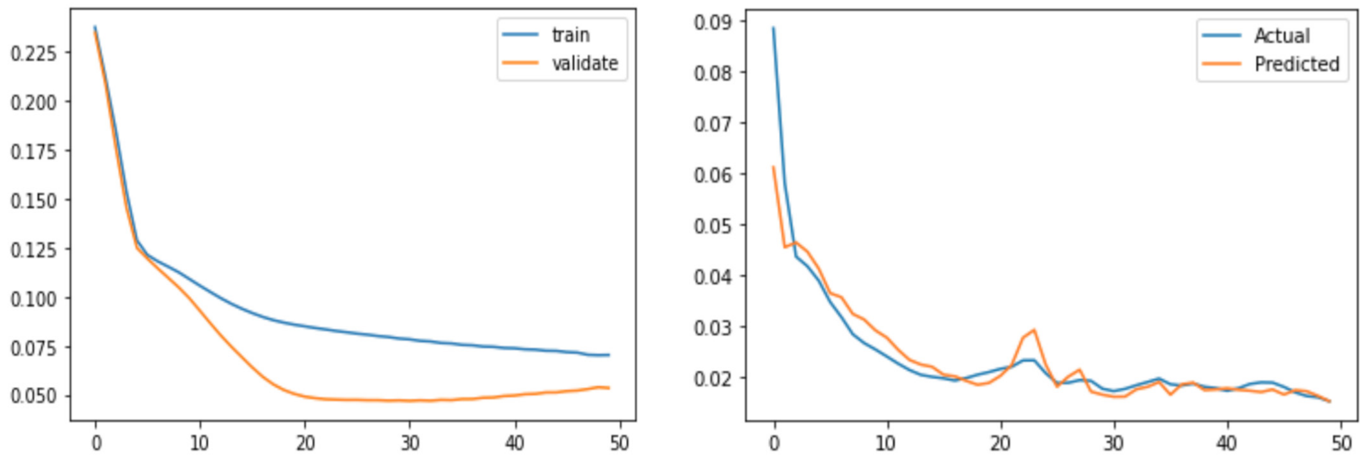


Figure 18. LSTM single lag (**left**) and multiple lag timestep (**right**) loss curve.

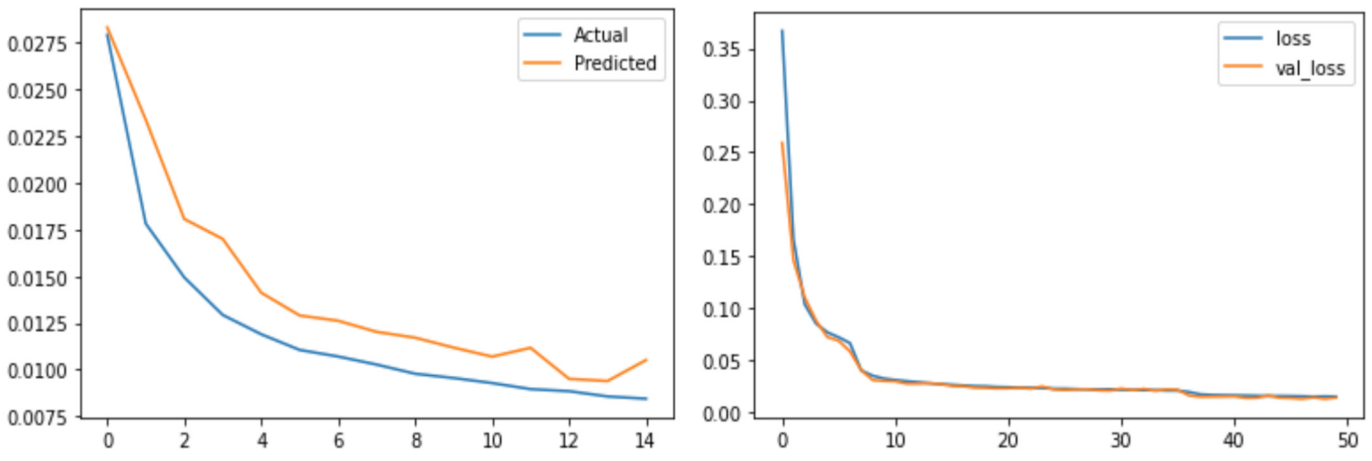


Figure 19. CNN-LSTM (**left**) and Bi-LSTM (**right**) loss curve.

Metrics MSE, RMSE, MAE, and MAPE were used to calculate prediction errors and evaluate the performance of the various models shown in the table below. Six models have been developed and the results were compared with the different error metrics mentioned in the table. The Decision Tree Regressor outperforms all other models for MSE, RMSE, and MAE error values. However, the MAPE of 0.556 is still higher than the standard accepted values between 0.05 and 0.09. The CNN-LSTM model was able to achieve the lowest error rate of 0.016 in MAPE. For the LSTM multivariate time series models, LSTM single lag performs inadequately with high error rates, while LSTM multiple lag performs better than single lag in all metrics, but the error rates for LSTM multiple lag are still higher than the standard accepted error values. The error rates for linear regression are lower in all metrics except MAPE when compared to models other than the Decision Tree Regressor. SVR (direct multi output regression) has a 0.079 value for MAE, which is higher than the Decision Tree Regressor and linear regression values. Bi-LSTM performed with high error rates for MSE and MAPE. Seeing the model performance, we also developed a stacked model with the CNN-LSTM and LSTM multi lag models, which achieved good results with an MSE value of 114.983, RMSE of 10.723, MAE of 5.010, and MAPE of 0.023. Table 7 provides the evaluation results for the stage 2 models, which were used for predicting the cattle pollutant.

Table 7. MSE, RMSE, MAE, MAPE for stage 2 modeling.

Models	MSE	RMSE	MAE	MAPE
Decision Tree Regressor	0.035	0.187	0.076	0.556
CNN-LSTM	59.870	7.738	3.511	0.016
LSTM—Single lag	55,527.062	767.944	233.430	-
—Multiple lag	272.517	20.356	10.263	0.045
Linear Regression	0.122	0.349	0.155	1.236
Bi-LSTM	428.907	20.710	3.920	3,875,151
Stacked	114.983	10.723	5.010	0.023

Table 8 summarizes the metrics values for Bi-LSTM. Comparing RMSE scores, SO_2 has a significantly lower value than that of other pollutants except for nitric oxide (NO), while $PM_{2.5}$ has the highest RMSE value followed by CO and then by PM_{10}. For MAE, SO_2 has the lowest value with 0.119, whereas $PM_{2.5}$ has the highest with 17.304. The MAPE scores are lowest for SO_2 at 0.089 and highest for NH_3 at 626,472,000. The highest MSE score is $PM_{2.5}$ at 3268.632 and the lowest is SO_2 with a score of 0.048. The MSLR has a high score of 0.052 for NH_3 and a low score of 0.001 for NO.

Table 8. MSE, RMSE, MAE, MAPE of Bi-LSTM for air pollutants.

Models	Pollutants	MSE	RMSE	MAE	MAPE	MSLR
Bi-LSTM	CO	103.541	10.175	6.033	2.992	0.015
	NH_3	2.293	1.514	1.362	6,264,720	0.052
	NO	0.472	0.687	0.211	34.711	0.001
	NO_2	21.333	4.618	3.236	3.726	0.003
	O_3	0.277	0.526	0.223	12,422.202	0.012
	PM_{10}	34.662	5.887	2.873	12.635	0.0264
	$PM_{2.5}$	3268.632	57.171	17.304	13.007	0.0342
	SO_2	0.048	0.220	0.119	0.089	0.007

Table 9 shows the modeling test results of machine learning models Decision Tree Regressor and linear regression for each of the pollutants considered for prediction. The relative mean squared error for each model was tabulated. Comparing error values for each pollutant, we found that the MSE error was the lowest for carbon monoxide (CO) and highest for ammonia (NH_3) for the Decision Tree and linear regression models. The errors calculated from both models are similar but linear regression is seen to perform better than the Decision Tree for all pollutants.

Table 9. MSE of Decision Tree and linear regression models for air pollutants.

Pollutants	Decision Tree	Linear Regression
CO	0.020	0.057
NH_3	1.474	1.882
NO	0.282	0.576
NO_2	0.035	0.085
O_3	0.298	0.771
PM_{10}	0.120	0.276
$PM_{2.5}$	0.158	0.356
SO_2	0.209	0.483

5. System Development

5.1. System Requirements Analysis

The targeted users of our system are cattle farms and farm owners, followed by the state's Department of Public Health and Environment, where the potential actors in this use case are the air quality inspectors. One other major use case that deals with the health of human beings is the homeowners living near livestock farms. Since this is a web application, the system can be accessed from anywhere by providing the location coordinates to get the pollutant levels.

Figure 20 provides the picturization of how the system interconnects every user with the use case. In the first use case, the owner uses the system to see the current pollution level based on the predefined air quality index with the current count of the cattle present in the farm. The web portal system has an option to enter the future count of cattle that are planned to be bought. The system now predicts the new pollutant level after the increase in the cattle count. This helps the cattle owner to plan and manage the number of cattle in their farm.

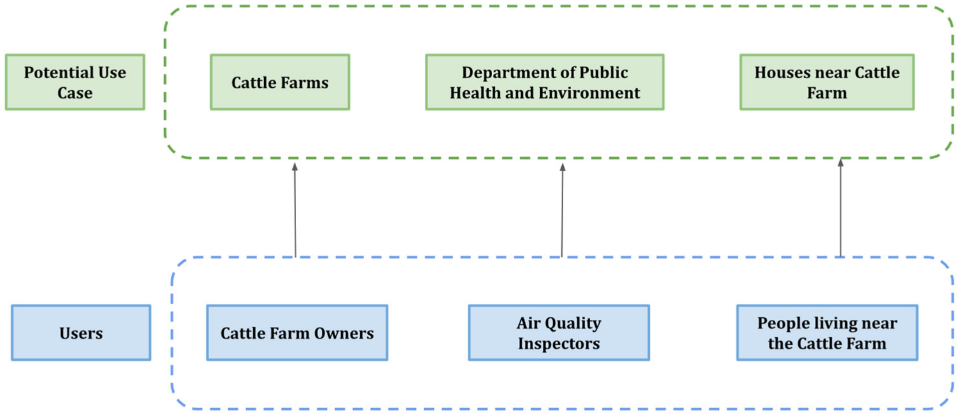


Figure 20. System boundary, actors, and use cases.

In the next use case, with the help of this system the air quality inspectors can measure the current level of pollutant in the air in their county. This helps them to potentially identify whether to provide or deny a cattle farm license to any new owner in the nearby coordinates. The important use case is that this system helps in preventing chronic obstructive pulmonary disease (COPD), since people living close to livestock farms are exposed to acute respiratory effects. So, this system which is used via their computers can help to predict the current pollutant level and the people living nearby the cattle farm can take precautionary measures well in advance.

5.2. System Design

For developing the cattle pollution detecting system, AWS was used for model training and storing the satellite data and weather data loaded from the weather API. The image detection and pollutant prediction deep learning (DL) [82] models were developed with Torch, Python3, and scikit-learn packages. All the different models developed were trained, validated, tested, and shared, and different stages were concatenated in Google Colab shared over a shared Google drive. All the models, training data, and results were continuously synced/uploaded with AWS. For training the model, satellite data were collected from USGS earth explorer, which provides a variety of satellite data out of which we collected raw data from Google Earth engine, which uses Sentinel.

Figure 21 shows the system architecture for the cattle pollution detecting system, highlighting the frontend and backend architecture. In the front end, the user provides the coordinates of their farm along with the cattle count to the 'Cattle Pollution Predictor' application interface via the computer monitor. There are two options: provide farm coordinates only or input cattle count along with the coordinates. In the backend, the user input goes into the cattle pollution detecting system, where it undergoes three stages: cattle counting, pollutant prediction, and visualization. In the first stage, the inputted coordinates are used to find the satellite images of the farm, which are then sent to YOLOv4, YOLOv5, ResNet, and RetinaNet models to find the number of cattle present in the farm. If the user has provided the cattle count in their input, a flag will skip this stage. After the cattle count is found, the cattle count and coordinate data go into the pollutant prediction system. These data, along with the corresponding weather data from the weather API, go through a stacked model prediction system containing a long short-term memory (LSTM) multivariate time series [83], CNN-LSTM [84], and Bi-LSTM [85] to predict the pollutant concentration for the following day. In the second stage, the predicted output pollutants are then classified based on severity. In the third stage, a visualization system is created to visualize the different classified pollutants from the previous stage showing the severity of each of the pollutant concentrations. The visualization results are sent to the Cattle Pollution Application and the user views the output via the computer monitor. The system provides the option to visualize the overall and individual severity of all pollutants for any hour from historical data and shows the trend for each of the different pollutant concentrations separately, as well as with the overall AQI. Figure 22 displays the user interface for the air pollutant severity map, while Figures 23 and 24 show the data visualizations for AQI and cattle count.

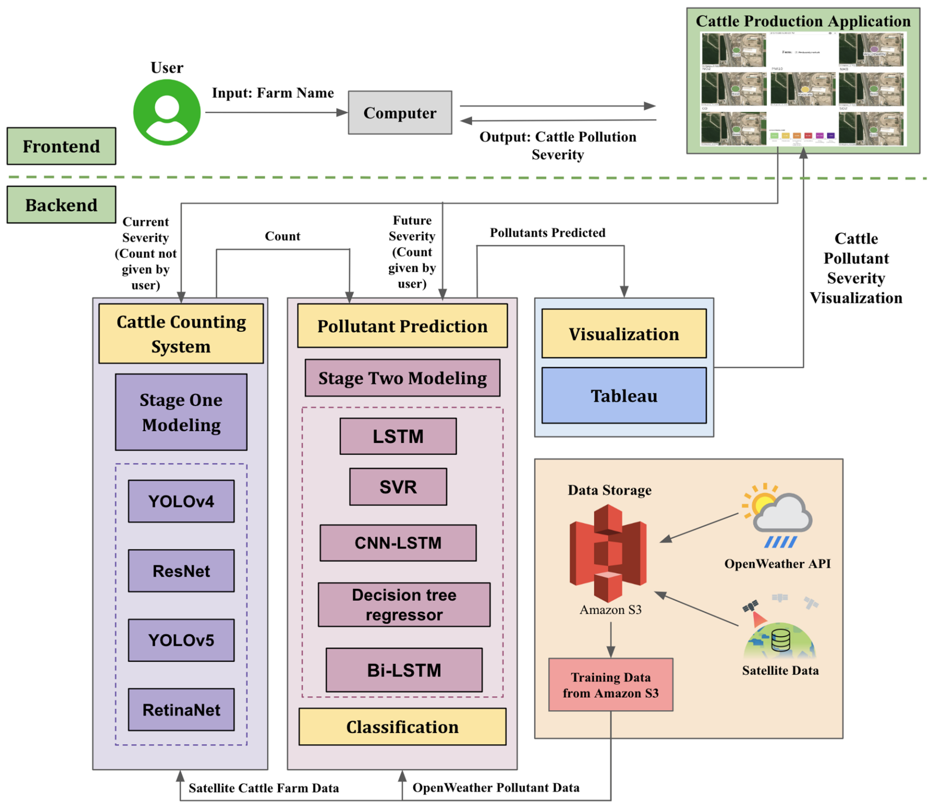
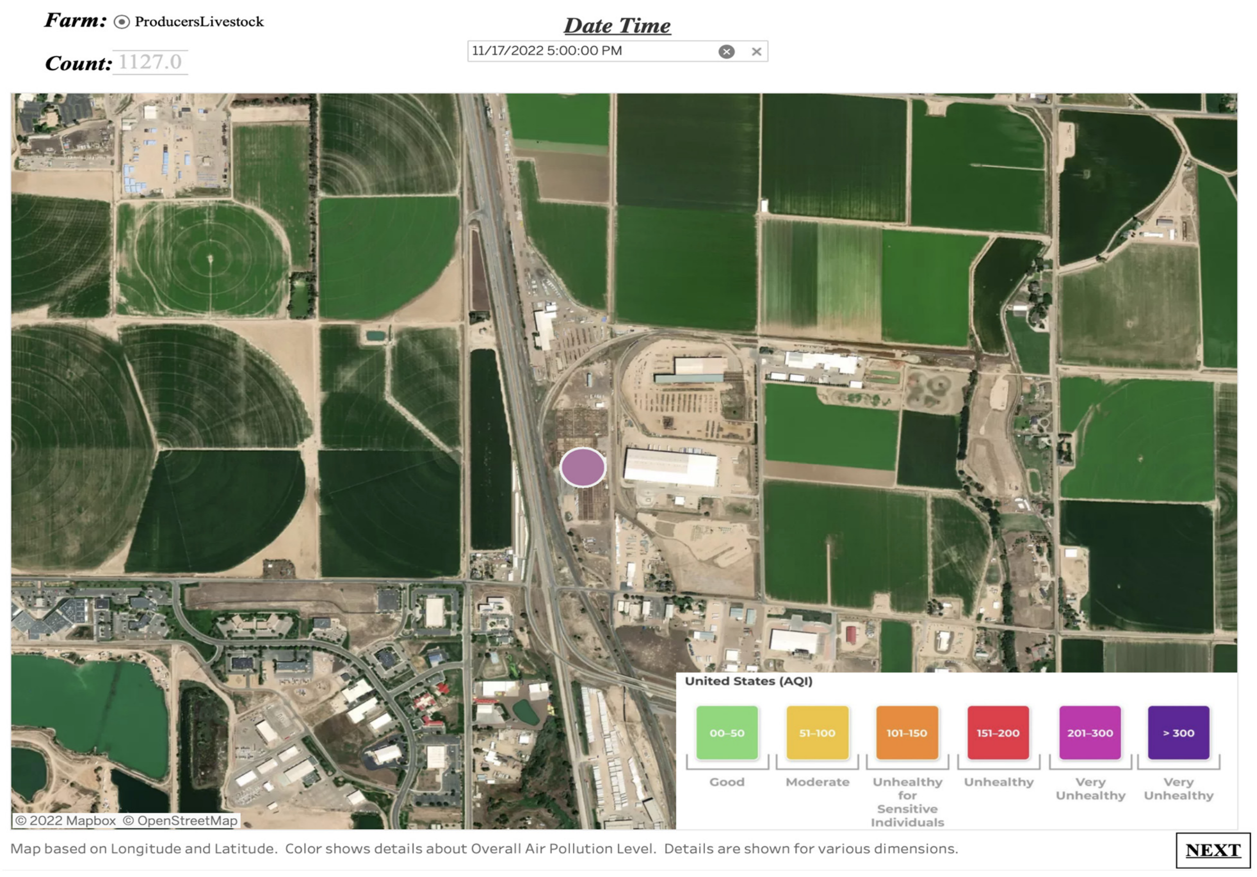
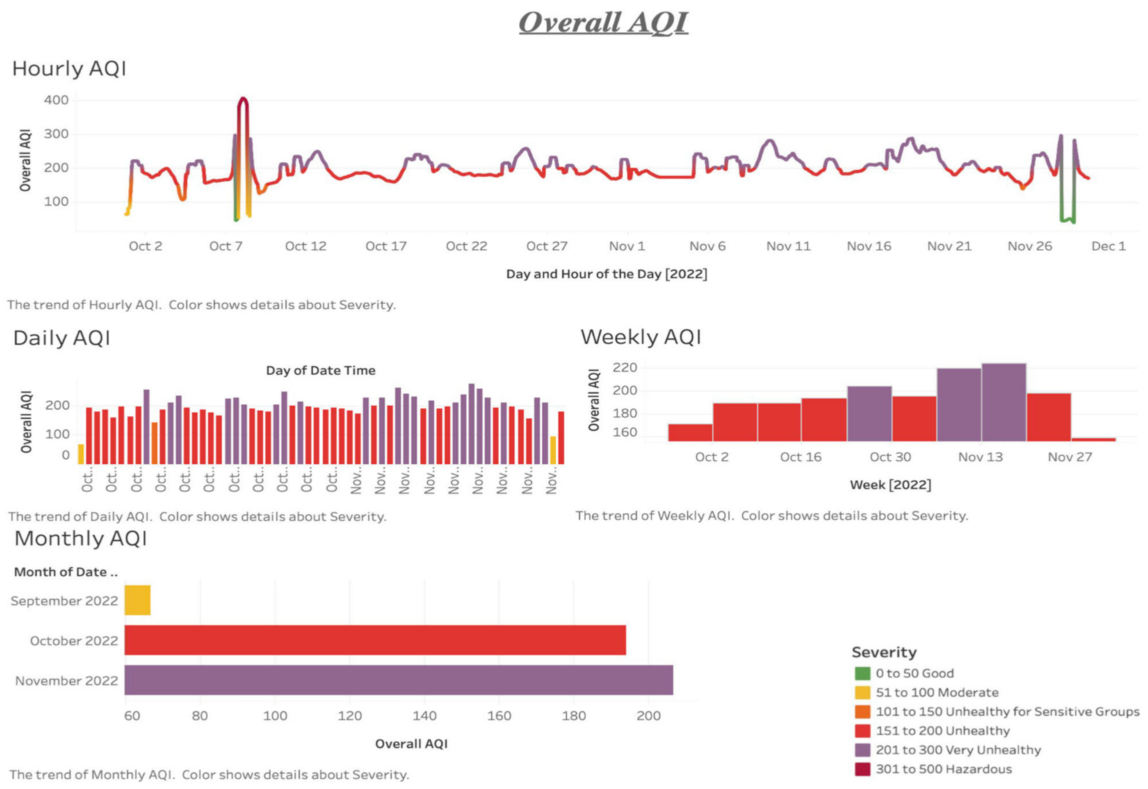


Figure 21. System frontend and backend architecture.



Map based on Longitude and Latitude. Color shows details about Overall Air Pollution Level. Details are shown for various dimensions.

Figure 22. User interface showing the color-coded pollutant severity map.



The trend of Hourly AQI. Color shows details about Severity.

The trend of Daily AQI. Color shows details about Severity.

The trend of Weekly AQI. Color shows details about Severity.

The trend of Monthly AQI. Color shows details about Severity.

Figure 23. User interface visualization showing overall AQI data.

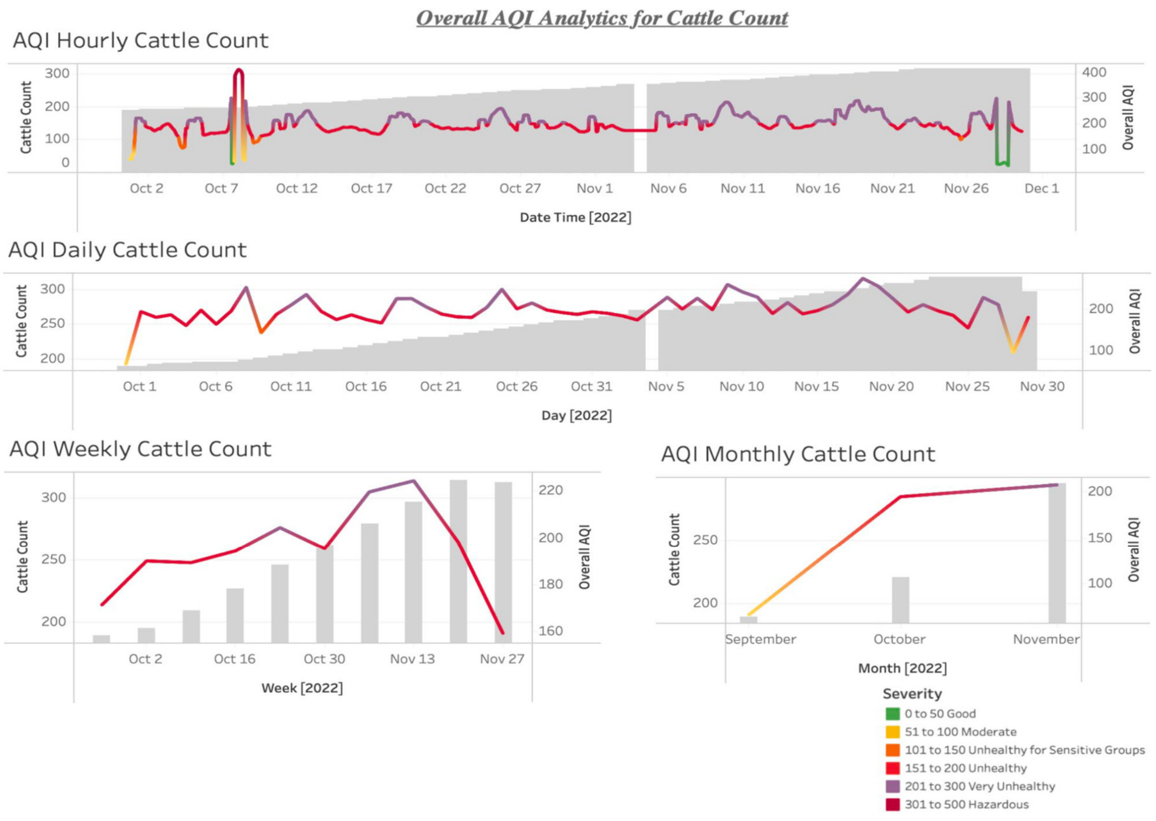


Figure 24. User interface visualization showing overall AQI for cattle count.

6. Conclusions

6.1. Summary

The objective of this project was to predict air quality in cattle farms with respect to the number of cattle present in each farm. In order to achieve this, the project was divided into three major stages. Firstly, we counted the number of cattle for which satellite images from the Google Earth engine were used in stage 1. We used four models: YOLOv5, YOLOv4, Detectron2, and RetinaNet. Our best results on the YOLOv5 model occurred with 650 epochs, with an average precision of 0.916 and recall of 0.912. YOLOv4 was able to produce an average precision of 0.872 and an average recall of 0.879 in 2000 iterations. Detectron2 provided an average precision of 0.871 and a recall of 0.875 in 3000 iterations. RetinaNet provided an average precision of 0.881 and an average recall of 0.887 with 300 epochs. In the second stage, four cattle farms from Colorado were chosen, namely Prado, Producers Livestock, Hillrose, and Docheff. The count of cattle from stage 1 models was given as the input to stage 2 models along with the air quality data from the OpenWeather API, which provided the live pollutant level of each location using location coordinates. The pollutants provided by the API were CO, NH_3, NO_2, O_3, $PM_{2.5}$, PM_{10}, and SO_2. Next, we created AQI and classified the level of pollutants. The models used for air quality prediction were LSTM (single lag and multi lag), Bi-LSTM, and CNN LSTM. The output of this stage was then fetched to the Tableau for the map visualization.

Lastly, a visualization was created in the Tableau map, along with the pollutant levels for each farm. The pollutant level was then mapped to the AQI created earlier based on distinct colors for each level of pollutant. A web application was developed using JS for the frontend and Flask API for the backend. The app was then linked to the saved models as a single Python file, which was connected to Tableau using GET and POST from Flask. Once a farm was chosen in the web app, the Flask API ran the Python file that performed the counting and prediction and visualized it in the Tableau. Finally, the interactive map dashboard was visualized to show the count and pollutant level for the seven pollutants in the cattle farms.

6.2. Recommendations for Future Work

As we targeted outdoor and open pen farms in this study, we assumed that there was no indoor manure processing or storing. Our study is not applicable to indoor storage collections and processing but, should there be different processing methods that may affect the correlation of the number of cattle with pollutant concentration, comparisons with indoor and outdoor farms can be performed using the same methodology for similar cattle counts to see the difference. Additionally, we plan to extend the scope of this study by further classifying pollutants from other livestock such as sheep, pigs, and ducks. Due to no funding, we had to limit our satellite images to free sources such as the Google Earth API, but we hope to obtain satellite images from a different satellite API in the future. Further improvements can be made to the model, and our model can be extended to predict the AQI of farms using user input of cattle count. This can be useful if the user plans to increase his livestock and wants to get an idea of AQI at a later stage in time.

Furthermore, this research can be extended to several states in the United States. Additional possibilities include other agricultural countries such as India and China, where the size and terrain of farms may differ from country to country. The guidelines for farm pollution standards and policies are different for each country and depend on their current government. These factors will make the model more robust and able to predict pollutants concentration for different farms globally.

Author Contributions: Investigation, J.H., R.J., R.R., C.P.S.M. and N.S.; Data curation, R.J., R.R., C.P.S.M. and N.S.; Formal analysis, R.J., R.R., C.P.S.M. and N.S.; Validation, J.H., R.J., R.R., C.P.S.M. and N.S.; Methodology, R.J., R.R., C.P.S.M. and N.S.; Visualization, J.H.; Writing—original draft, J.H.; Writing—review & editing, J.H. and J.W.; Project administration, J.G.; Supervision, J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here in the references listed: [45–49].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lutz, W.; KC, S. Dimensions of global population projections: What do we know about future population trends and structures? *Philos. Trans. R. Soc. B Biol. Sci.* **2010**, *365*, 2779–2791. [CrossRef] [PubMed]
2. Greenwood, P.L. An overview of beef production from pasture and feedlot globally, as demand for beef and the need for sustainable practices increase. *Animal* **2021**, *15*, 100295. [CrossRef]
3. Bogaerts, M.; Cirhigiri, L.; Robinson, I.; Rodkin, M.; Hajjar, R.; Junior, C.C.; Newton, P. Climate change mitigation through intensified pasture management: Estimating greenhouse gas emissions on cattle farms in the Brazilian Amazon. *J. Clean. Prod.* **2017**, *162*, 1539–1550. [CrossRef]
4. Thumba, D.; Lazarova-Molnar, S.; Niloofar, P. Estimating Livestock Greenhouse Gas Emissions: Existing Models, Emerging Technologies and Associated Challenges. In Proceedings of the 2021 6th International Conference on Smart and Sustainable Technologies (SpliTech), Bol and Split, Croatia, 8–11 September 2021. [CrossRef]
5. Castelli, M.; Clemente, F.; Popovič, A.; Silva, S.; Vanneschi, L. A Machine Learning Approach to Predict Air Quality in California. *Complexity* **2020**, *2020*, 8049504. [CrossRef]
6. Borlée, F.; Yzermans, C.J.; Aalders, B.; Rooijackers, J.; Krop, E.; Maassen, C.B.M.; Schellevis, F.; Brunekreef, B.; Heederik, D.; Smit, L.A.M.; et al. Air Pollution from Livestock Farms Is Associated with Airway Obstruction in Neighboring Residents. *Am. J. Respir. Crit. Care Med.* **2017**, *196*, 1152–1161. [CrossRef] [PubMed]
7. Domingo, N.; Balasubramanian, S.; Thakrar, S.K.; Clark, M.A.; Adams, P.J.; Marshall, J.D.; Muller, N.Z.; Pandis, S.N.; Polasky, S.; Robinson, A.L.; et al. Air quality–related health damages of food. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2013637118. [CrossRef] [PubMed]
8. Madhuri, V.M.; Samyama, G.G.H.; Kamalapurkar, S. Air Pollution Prediction Using Machine Learning Supervised Learning Approach. *Int. J. Sci. Technol. Res.* **2020**, *9*, 118–123.
9. Knight, C.H. Sensor techniques in ruminants: More than fitness trackers. *Animal* **2020**, *14*, s187–s195. [CrossRef]

10. Cruz-Rivero, L.; Mateo-Díaz, N.F.; Purroy-Vásquez, R.; Angeles-Herrera, D.; Cruz, F.O. Statistical analysis for a non-invasive methane gas and carbon dioxide measurer for ruminants. In Proceedings of the 2020 IEEE International Conference on Engineering Veracruz (ICEV), Boca del Rio, Mexico, 26–29 October 2020; Available online: https://ieeexplore.ieee.org/document/9289668 (accessed on 11 October 2022).
11. Joharestani, M.Z.; Cao, C.; Ni, X.; Bashir, B.; Talebiesfandarani, S. PM2.5 Prediction Based on Random Forest, XGBoost, and Deep Learning Using Multisource Remote Sensing Data. *Atmosphere* **2019**, *10*, 373. [CrossRef]
12. Holloway, T.; Miller, D.; Anenberg, S.; Diao, M.; Duncan, B.; Fiore, A.M.; Henze, D.K.; Hess, J.; Kinney, P.L.; Liu, Y.; et al. Satellite Monitoring for Air Quality and Health. *Annu. Rev. Biomed. Data Sci.* **2021**, *4*, 417–447. [CrossRef]
13. Lakshmi, V.S.; Vijaya, M.S. A Study on Machine Learning-Based Approaches for PM2.5 Prediction. *Sustain. Commun. Netw. Appl.* **2022**, *93*, 163–175. [CrossRef]
14. Weather API—OpenWeatherMap. 2022. Available online: https://openweathermap.org/api (accessed on 29 September 2022).
15. Wikipedia Contributors. Air Quality Index. *Wikipedia*, 27 April 2019. Available online: https://en.wikipedia.org/wiki/Air_Quality_Index (accessed on 29 September 2022).
16. Yang, W.; Deng, M.; Xu, F.; Wang, H. Prediction of hourly PM2.5 using a space-time support vector regression model. *Atmos. Environ.* **2018**, *181*, 12–19. [CrossRef]
17. Aini, N.; Mustafa, M. Data Mining Approach to Predict Air Pollution in Makassar. In Proceedings of the 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS), Manado, Indonesia, 27–28 October 2020. [CrossRef]
18. Delavar, M.R.; Gholami, A.; Shiran, G.R.; Rashidi, Y.; Nakhaeizadeh, G.R.; Fedra, K.; Afshar, S.H. A Novel Method for Improving Air Pollution Prediction Based on Machine Learning Approaches: A Case Study Applied to the Capital City of Tehran. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 99. [CrossRef]
19. Sousa, V.; Backes, A. Unsupervised Segmentation of Cattle Images Using Deep Learning. In Proceedings of the Anais do XVII Workshop de Visão Computacional (WVC 2021), Virtual, 22–23 November 2021. [CrossRef]
20. Chen, J.; de Hoogh, K.; Gulliver, J.; Hoffmann, B.; Hertel, O.; Ketzel, M.; Bauwelinck, M.; van Donkelaar, A.; Hvidtfeldt, U.A.; Katsouyanni, K.; et al. A comparison of linear regression, regularization, and machine learning algorithms to develop Europe-wide spatial models of fine particles and nitrogen dioxide. *Environ. Int.* **2019**, *130*, 104934. [CrossRef]
21. Ahmad, T.; Ma, Y.; Yahya, M.; Ahmad, B.; Nazir, S.; Haq, A. Object Detection through Modified YOLO Neural Network. *Sci. Program.* **2020**, *2020*, 8403262. [CrossRef]
22. Yi, H.; Xiong, Q.; Zou, Q.; Xu, R.; Wang, K.; Gao, M. A Novel Random Forest and its Application on Classification of Air Quality. In Proceedings of the 2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI), Toyama, Japan, 7–11 July 2019. [CrossRef]
23. Ma, J.; Cheng, J.; Xu, Z.; Chen, K.; Lin, C.; Jiang, F. Identification of the most influential areas for air pollution control using XGBoost and Grid Importance Rank. *J. Clean. Prod.* **2020**, *274*, 122835. [CrossRef]
24. Cao, Y.; Zhang, W.; Bai, X.; Chen, K. Detection of excavated areas in high-resolution remote sensing imagery using combined hierarchical spatial pyramid pooling and VGGNet. *Remote Sens. Lett.* **2021**, *12*, 1269–1280. [CrossRef]
25. Haifley, T. Linear logistic regression: An introduction. In Proceedings of the IEEE International Integrated Reliability Workshop Final Report, Lake Tahoe, CA, USA, 21–24 October 2002; pp. 184–187. [CrossRef]
26. Liu, T.; Wu, T.; Wang, M.; Fu, M.; Kang, J.; Zhang, H. Recurrent Neural Networks based on LSTM for Predicting Geomagnetic Field. In Proceedings of the 2018 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), Bali, Indonesia, 20–21 September 2018; pp. 1–5. [CrossRef]
27. Tsang, S.-H. Review: RetinaNet—Focal Loss (Object Detection). Medium, 26 March 2019. Available online: https://towardsdatascience.com/review-retinanet-focal-loss-object-detection38fba6afabe4#:~:text=Using%20larger%20scales%20allows%20RetinaNet (accessed on 1 December 2022).
28. Xu, Y.; Ho, H.C.; Wong, M.S.; Deng, C.; Shi, Y.; Chan, T.-C.; Knudby, A. Evaluation of machine learning techniques with multiple remote sensing datasets in estimating monthly concentrations of ground-level $PM_{2.5}$. *Environ. Pollut.* **2018**, *242*, 1417–1426. [CrossRef]
29. Singh, J.K.; Goel, A.K. Prediction of Air Pollution by using Machine Learning Algorithm. In Proceedings of the 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 19–20 March 2021. [CrossRef]
30. Liu, B.; Xu, Z.; Kang, Y.; Cao, Y.; Pei, L. Air Pollution Lidar Signals Classification Based on Machine Learning Methods. In Proceedings of the 2020 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020; pp. 13–18. [CrossRef]
31. Hempel, S.; Adolphs, J.; Landwehr, N.; Willink, D.; Janke, D.; Amon, T. Supervised Machine Learning to Assess Methane Emissions of a Dairy Building with Natural Ventilation. *Appl. Sci.* **2020**, *10*, 6938. [CrossRef]
32. Reed, K.; Moraes, L.; Casper, D.; Kebreab, E. Predicting nitrogen excretion from cattle. *J. Dairy Sci.* **2015**, *98*, 3025–3035. [CrossRef]
33. Hempel, S.; Adolphs, J.; Landwehr, N.; Janke, D.; Amon, T. How the Selection of Training Data and Modeling Approach Affects the Estimation of Ammonia Emissions from a Naturally Ventilated Dairy Barn—Classical Statistics versus Machine Learning. *Sustainability* **2020**, *12*, 1030. [CrossRef]
34. Kanniah, K.D.; Zaman, N.A.F.K. Exploring the Link Between Ground Based PM2.5 and Remotely Sensed Aerosols and Gases Data to Map Fine Particulate Matters in Malaysia Using Machine Learning Algorithms. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 7216–7219. [CrossRef]

35. Hadjimitsis, D.G. Aerosol optical thickness (AOT) retrieval over land using satellite image based algorithm. *Air Qual. Atmos. Health* **2009**, *2*, 89–97. [CrossRef]
36. Stirnberg, R.; Cermak, J.; Fuchs, J.; Andersen, H. Mapping and Understanding Patterns of Air Quality Using Satellite Data and Machine Learning. *J. Geophys. Res. Atmos.* **2020**, *125*, e2019JD031380. [CrossRef]
37. Liao, Q.; Zhu, M.; Wu, L.; Pan, X.; Tang, X.; Wang, Z. Deep Learning for Air Quality Forecasts: A Review. *Curr. Pollut. Rep.* **2022**, *6*, 399–409. [CrossRef]
38. Haq, M.A. Smotednn: A novel model for air pollution forecasting and aqi classification. *Comput. Mater. Contin.* **2022**, *71*, 1403–1425.
39. Shakeel, A.; Waqas, S.; Ali, M. Deep built-structure counting in satellite imagery using attention based re-weighting. *ISPRS J. Photogramm. Remote Sens.* **2019**, *151*, 313–321. [CrossRef]
40. de Vasconcellos, B.C.; Trindade, J.P.; da Silva Volk, L.B.; de Pinho, L.B. Method Applied To Animal MonitoringThrough VANT Images. *IEEE Lat. Am. Trans.* **2020**, *18*, 1280–1287. [CrossRef]
41. Soares, V.; Ponti, M.; Gonçalves, R.; Campello, R. Cattle counting in the wild with geolocated aerial images in large pasture areas. *Comput. Electron. Agric.* **2021**, *189*, 106354. [CrossRef]
42. Laradji, I.; Rodriguez, P.; Kalaitzis, F.; Vazquez, D.; Young, R.; Davey, E.; Lacoste, A. Counting Cows: Tracking Illegal Cattle Ranching from High-Resolution Satellite Imagery. *arXiv* **2022**. [CrossRef]
43. Gołasa, P.; Wysokiński, M.; Bieńkowska-Gołasa, W.; Gradziuk, P.; Golonko, M.; Gradziuk, B.; Siedlecka, A.; Gromada, A. Sources of Greenhouse Gas Emissions in Agriculture, with Particular Emphasis on Emissions from Energy Used. *Energies* **2021**, *14*, 3784. [CrossRef]
44. Amazon EC2 G4 Instances—Amazon Web Services (AWS). Amazon Web Services, Inc. Available online: https://aws.amazon.com/ec2/instance-types/g4/ (accessed on 29 September 2022).
45. Satellite Data: What Spatial Resolution Is Enough for You? Eos.com. 2019. Available online: https://eos.com/blog/satellite-data-what-spatial-resolution-is-enough-for-you/ (accessed on 29 September 2022).
46. Landsat 8 Datasets in Earth Engine | Earth Engine Data Catalog. Google Developers. Available online: https://developers.google.com/earth-engine/datasets/catalog/landsat-8 (accessed on 29 September 2022).
47. Air Pollution—OpenWeatherMap. Available online: https://openweathermap.org/api/air-pollution (accessed on 29 September 2022).
48. NAIP: National Agriculture Imagery Program | Earth Engine Data Catalog. Google Developers. Available online: https://developers.google.com/earthengine/datasets/catalog/USDA_NAIP_DOQQ#citations (accessed on 29 September 2022).
49. Nitrogen, Ammonia Emissions and the Dairy Cow. Penn State Extension. Available online: https://extension.psu.edu/nitrogen-ammonia-emissions-and-the-dairy-cow (accessed on 29 September 2022).
50. AQI Calculation Update. Airveda. Available online: https://www.airveda.com/blog/AQI-calculation-update (accessed on 29 September 2022).
51. Roboflow: Give Your Software the Power to See Objects in Images and Video. Roboflow. Available online: https://roboflow.com/ (accessed on 29 September 2022).
52. Tableau. Tableau: Business Intelligence and Analytics Software. Tableau Software. 2018. Available online: https://www.tableau.com/ (accessed on 29 September 2022).
53. Yu, J.; Zhang, W. Face Mask Wearing Detection Algorithm Based on Improved YOLO v4. *Sensors* **2021**, *21*, 3263. [CrossRef] [PubMed]
54. Supeshala, C. YOLO v4 or YOLO v5 or PP-YOLO? Medium, 23 August 2020. Available online: https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109 (accessed on 29 September 2022).
55. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
56. Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. Automatic Ship Detection Based on RetinaNet Using Multi-Resolution Gaofen-3 Imagery. *Remote Sens.* **2019**, *11*, 531. [CrossRef]
57. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
58. Pham, V.; Pham, C.; Dang, T. Road Damage Detection and Classification with Detectron2 and Faster R-CNN. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020. [CrossRef]
59. Meta | Social Metaverse Company. Available online: https://about.meta.com/ (accessed on 29 September 2022).
60. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
61. Aalen, O.O. A linear regression model for the analysis of life times. *Stat. Med.* **1989**, *8*, 907–925. [CrossRef]
62. Train Convolutional Neural Network for Regression—MATLAB & Simulink. MathWorks. Available online: https://www.mathworks.com/help/deeplearning/ug/train-aconvolutional-neural-network-for-regression.html (accessed on 21 October 2022).
63. Jung, H.; Kim, B.; Lee, I.; Yoo, M.; Lee, J.; Ham, S.; Woo, O.; Kang, J. Detection of masses in mammograms using a one-stage object detector based on a deep convolutional neural network. *PLoS ONE* **2018**, *13*, e0203355. [CrossRef]
64. Tasdelen, A.; Sen, B. A hybrid CNN-LSTM model for pre-miRNA classification. *Sci. Rep.* **2021**, *11*, 14125. [CrossRef]

65. Wattal, K.; Singh, S.K. Multivariate Air Pollution Levels Forecasting. In Proceedings of the 2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS), Ernakulam, India, 2–4 September 2021; pp. 165–169. [CrossRef]
66. Chauhan, R.; Kaur, H.; Alankar, B. Air Quality Forecast using Convolutional Neural Network for Sustainable Development in Urban Environments. *Sustain. Cities Soc.* **2021**, *75*, 103239. [CrossRef]
67. Zhang, M.; Wu, D.; Xue, R. Hourly prediction of PM2.5 concentration in Beijing based on Bi-LSTM neural network. *Multimed. Tools Appl.* **2021**, *80*, 24455–24468. [CrossRef]
68. Shang, W.; Huang, H.; Zhu, H.; Lin, Y.; Wang, Z.; Qu, Y. An Improved kNN Algorithm—Fuzzy Knn. *Comput. Intell. Secur.* **2005**, *3801*, 741–746. [CrossRef]
69. Team, K. Keras Documentation: Object Detection with RetinaNet. keras.io. Available online: https://keras.io/examples/vision/retinanet/ (accessed on 29 September 2022).
70. ultralytics/yolov5. GitHub. 14 May 2022. Available online: https://github.com/ultralytics/yolov5/ (accessed on 14 May 2022).
71. Sabina, N.; Aneesa, M.P.; Haseena, P.V. Object Detection Using YOLO and Mobilenet SSD: A Comparative Study. *Int. J. Eng. Res. Technol.* **2022**, *11*, 134–138.
72. San Jose State University Library. Available online: https://ieeexplore-ieeeorg.libaccess.sjlibrary.org/stamp/stamp.jsp?tp=&arnumber=9615648&tag=1 (accessed on 14 May 2022).
73. Mean Squared Error: Definition and Example. Statistics How to. 2022. Available online: https://www.statisticshowto.com/probability-and-statistics/statisticsdefinitions/mean-squared-error/ (accessed on 21 October 2022).
74. de Myttenaere, A.; Golden, B.; Le Grand, B.; Rossi, F. Mean absolute percentage error for regression models. *Neurocomputing* **2016**, *192*, 38–48. [CrossRef]
75. Md Atiqur, R.; Wang, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International Symposium on Visual Computing*; Springer: Cham, Switzerland, 2016.
76. Measuring Performance: The Confusion Matrix. Medium. 2022. Available online: https://towardsdatascience.com/measuring-performance-the-confusion-matrix25c17b78e516 (accessed on 29 September 2022).
77. The F1 Score. Medium. 2022. Available online: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6 (accessed on 29 September 2022).
78. Google. Classification: ROC Curve and AUC | Machine Learning Crash Course. Google Developers. 2019. Available online: https://developers.google.com/machine-learning/crashcourse/classification/roc-and-auc (accessed on 29 September 2022).
79. Narkhede, S. Understanding AUC—ROC Curve. Medium, 26 June 2018. Available online: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 (accessed on 29 September 2022).
80. van Kersen, W.; Oldenwening, M.; Aalders, B.; Bloemsma, L.D.; Borlée, F.; Heederik, D.; Smit, L.A. Acute respiratory effects of livestock-related air pollution in a panel of COPD patients. *Environ. Int.* **2020**, *136*, 105426. [CrossRef] [PubMed]
81. Drusch, M.; Del Bello, U.; Carlier, S.; Colin, O.; Fernandez, V.; Gascon, F.; Hoersch, B.; Isola, C.; Laberinti, P.; Martimort, P.; et al. Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sens. Environ.* **2012**, *120*, 25–36. [CrossRef]
82. Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
83. Shahid, F.; Zameer, A.; Muneeb, M. Predictions for COVID-19 with Deep Learning Models of LSTM, GRU and Bi-LSTM. *Chaos Solitons Fractals* **2020**, *140*, 110212. [CrossRef]
84. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
85. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [CrossRef]