Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

Deep learning detection of types of water-bodies using optical variables and ensembling

Nida Nasir [a,*], Afreen Kansal [b], Omar Alshaltone [a,c], Feras Barneih [a,c], Abdallah Shanableh [a,c], Mohammad Al-Shabi [a,c], Ahmed Al Shammaa [c,d]

[a] Research Institute of Science and Engineering, University of Sharjah, Sharjah, United Arab Emirates
[b] Department of Statistics, London School of Economics, London, United Kingdom
[c] College of Engineering, University of Sharjah, Sharjah, United Arab Emirates
[d] University of Khorfakkan, Sharjah, United Arab Emirates

ARTICLE INFO

Keywords:
Stacked modeling
ANOVA
Smote
Classification
Meta learning

ABSTRACT

Water features are one of the most crucial environmental elements for strengthening climate-change adaptation. Remote sensing (RS) technologies driven by artificial intelligence (AI) have emerged as one of the most sought-after approaches for automating water information extraction and indeed. In this paper, a stacked ensemble model approach is proposed on AquaSat dataset (more than 500,000 images collection via satellite and Google Earth Engine). A one-way Analysis of variance (ANOVA) test and the Kruskal Wallis test are conducted for various optical-based variables at 99% significance level to understand how these vary for different water bodies. An oversampling is done on the training data using Synthetic Minority Oversampling Technique (SMOTE) to solve the problem of class imbalance while the model is tested on an imbalanced data, replicating the real-life situation. To enhance state-of-the-art, the pros of standalone machine learning classifiers and neural networks have been utilized. The stacked model obtained 100% accuracy on the testing data when using the decision tree classifier as the meta model. This study has been cross validated five-fold and will help researchers working in in-situ water bodies detection with the use of stacked model classification.

1. Introduction

Inland water quality estimates from satellites have the potential to improve our ability to observe and track the behavior of enormous bodies of water. In areas with sparse or no data, researchers can supplement in-situ sampling with satellite remote sensing to learn about water quality (Alshaltone et al., 2021). Researchers in the fields of limnology, oceanography, and hydrology have been eager to develop standardized methods for gleaning information about water quality from remotely sensed images ever since the first Landsat satellites were launched. By training and validating on larger data sets of coincident field and satellite observations, remote sensing models of water quality may be improved. To better incorporate remote sensing into studies of inland waters and aid in the development of models. When compared to ocean remote sensing, which benefits from large, open, robust data sets created to combine in situ and radiometric observations with satellite data, progress in inland water remote sensing appears glacial. Therefore, more general-purpose algorithms and methods can be developed rapidly (Blondeau-Patissier et al., 2014). The development of methods for rivers, lakes, and the near-shore environment is further complicated by the greater optical complexity of inland waters, where spectral signatures reflect a mixture of inorganic suspended sediment, organic suspended sediment, algae, dissolved organic matter, and other constituents (Mishra et al., 2017).

This section is where the authors delve even deeper into the methods for detecting bodies of water using artificial intelligence. Most recent methods for water body detection made use of deeper neural networks, though some studies relied solely on ML techniques (e.g., RF and SVM). In order to assess multiband RS data for water body extraction in the Himalayas, the researchers used band techniques (with slope, NDVI, and NDWI added as three additional bands to integrate more information into ML training), and then applied a support vector machine (SVM), a decision tree (DT), and a random forest (RF) (Acharya et al., 2019). Their models did very well on both flat and hilly terrain, but they had to

* Corresponding author.
E-mail addresses: nnasir@sharjah.ac.ae (N. Nasir), a.kansal1@lse.ac.uk (A. Kansal), oalshaltone@sharjah.ac.ae (O. Alshaltone), fbarneih@sharjah.ac.ae (F. Barneih), shanableh@sharjah.ac.ae (A. Shanableh), malshabi@sharjah.ac.ae (M. Al-Shabi), alshammaa@sharjah.ac.ae (A. Al Shammaa).

https://doi.org/10.1016/j.iswa.2023.200222
Received 14 November 2022; Received in revised form 11 January 2023; Accepted 30 March 2023
Available online 12 April 2023

use this method to differentiate between snow and very high altitudes (which requires more preprocessing and places where their method cannot be used with optical data). The authors ran numerous experiments to compare the performance of different input bands (NDWI versus specific input bands from Landsat data), but they were only able to do so visually. The authors found that, with the exception of NNs, it's better to add a single secondary band than a large number of them to most ML techniques. The health of humans and other animals, in addition to the state of the environment and the economy, could be severely compromised by algal blooms. Many factors can lead to algal blooms, making it difficult and time-consuming to gather the data needed to predict their occurrence. ML models can provide early warning for these events by incorporating time series data on fundamental factors governing water quality. Reservoirs can be created by damming a number of rivers, and in order to predict harmful algal blooms, a linear regression model was compared to an MLP, RNN, and LSTM (Lee & Lee, 2018).

In Wang et al. (2019), the authors focused solely on one-dimensional inputs and outputs, which was a significant accomplishment in itself (i.e., a 1D time series of dissolved oxygen as an input to predict dissolved oxygen at some time in the future). The results were encouraging, but the authors did note that training on multiple time series simultaneously could improve the architecture. According to the scientists' findings, projections made for a horizon of six months or more tend to be inaccurate. It's not enough to simply monitor water for different contaminant levels; once pollution is found, it must be traced back to its original source. Using cross-correlation, contaminants were linked to a variety of water quality indicators.

They used an LSTM to correlate pollutants with nearby businesses using the strongly linked water quality metrics. RNNs, like LSTMs, have proven to be accurate at predicting time series, but they are often criticized for being difficult to understand. However, it is difficult for ecological models based on deterministic processes to accurately depict trends over extended periods of time. Forecasts of lake phosphorus levels were improved by combining a process-based model with a recurrent neural network (RNN) to remove outlier predictions and improve alignment. The predictions made by NNs are more in line with ecological principles when their output is constrained using physics-based models (Hanson et al., 2020).

Increased protein loads in water sources can cause harmful algal blooms, which in turn can cause eutrophication. Due to the potential for this method to create dead zones, animals would be wiped out and the economy would suffer. Therefore, it is important to track chlorophyll-a levels in water to anticipate algal blooms.

To address this gap, Zhao et al. (2021) compared DL models to traditional ML and curve-fitting methods for making predictions about chlorophyll-a concentrations based on time series data and RS images. Since the authors only studied a single lake, their data is limited. And so it was that DL models generally failed to deliver satisfactory results. In addition, the ML models used here needed more information and processing power to succeed than less complex models would have. Constraints on resolution and the presence of background noise make it difficult to detect changes in water quality in inland bodies of water.

Using a proximal hyperspectral imager and high spectral and temporal time series data, the authors in Sun et al. (2022) took continuous measurements for water body detection. The research showed that index-based approaches to water quality monitoring were difficult to calibrate due to the arbitrary nature of thresholding values, while ML and DL models performed much better. However, the authors demonstrate that their models do not transfer well to other bodies of water with different distributions of water quality parameters.

Study Tambe et al. (2021) proposed a novel convolution-inception block in a network called W-Net to identify bodies of water in RS images. W-Net requires less processing time than other CNN models because it only needs to train on a smaller number of photos to reliably extract water bodies, and its authors pointed out that this is all thanks to the model's clever use of inception layers. Although W-Net was ultimately chosen over competing CNN designs, the authors still had to manually annotate hundreds of images.

Additionally, Dang and Li (2021) researchers created an original MSResNet that was trained on a massive collection of unlabeled RS images. In addition to its unsupervised water body extraction capabilities, MSResNet can also tell the difference between water bodies of different resolutions and shapes. But their system cannot tell the difference between water, farmland, and desert.

The paper "A novel method for improving the robustness of deep learning-based malware detectors against adversarial attacks" (Shaukat et al., 2022) presents a new approach for increasing the resilience of deep learning-based malware detection systems against adversarial attacks. The proposed method is a combination of data augmentation techniques and a regularization method known as "adversarial training" to improve the robustness of deep learning-based malware detectors. The authors of the paper evaluated the effectiveness of their proposed method using a dataset of malware samples, and the results showed that their approach leads to significant improvements in the robustness of deep learning-based malware detectors against adversarial attacks.

1.1. *Addressing research gap*

The main aspects of this proposed study are as follows:

- Extensive Dataset: AquaSat dataset (more than 500,000 images collection via satellite and Google Earth Engine.
- Optical Variables Analysis: A one-way ANOVA test and the Kruskal-Wallis test are conducted for 20 optical-based variables while calculating their p-value.
- Sampling: An oversampling is done on the training data using SMOTE to solve the problem of class imbalance while the model is tested on imbalanced data, recreating the real-life scenario.
- Classification (Machine learning (ML)): Eight Classifiers are employed, using the Adam optimizer, Categorical Cross-entropy loss function and accuracy as the metric. Including Early Stopping and Model Checkpoint callbacks.
- Ensemble Meta-Learning (Deep Learning (DL)): The stacked model obtained 100% accuracy on the testing data when using the decision tree classifier as the meta-model.
- Cross validation-Five fold cross validation has been done and discussed for deep learning ensemble method.

The rest of the paper is structured as follows. Section 2 presents the Literature review. Section 3 Methodology explaining the dataset and ML/DL models, along with the proposed model. Section 4 provides all simulated and calculated results. Section 5 discusses all aspects of the study, including the future. Section 6 concludes the paper and outlines future research.

2. Literature review

Authors in Javed et al. (2021) review the two main types of recommendation systems: content-based and context-based. Content-based recommendation systems use information about the characteristics of an item to recommend similar items to a user, while context-based recommendation systems use information about the user's current context to make recommendations. Both types of recommendation systems have their own strengths and limitations, and the choice between them depends on the specific goals and needs of the system being developed.

Deep learning, a type of machine learning that involves training artificial neural networks on large datasets, can be used to improve the accuracy of recommendation systems. Stacked modeling, which involves training multiple models and combining their predictions, can be useful when working with complex or diverse datasets. Also, statistical techniques such as ANOVA (analysis of variance) and SMOTE (synthetic minority oversampling technique) can also be used to improve the performance of recommendation systems. ANOVA can be used to identify significant differences between groups, while SMOTE can be used to address imbalanced datasets, where one class is underrepresented. On the other hand, classification and meta learning algorithms may also be useful in certain situations, depending on the characteristics of the data and the type of recommendations being made. In summary, the authors review the various techniques and approaches that can be used to develop effective recommendation systems.

2.1. Dataset background

The resources required to remotely assess inland water quality have increased dramatically during the last decade (Nasir et al., 2022). The switch to open access Landsat data in 2008 is one instance that is mentioned, as a result, publications increased and studies' scope and length greatly expanded. The Landsat archive, however, is just one of several petabyte-sized archives of earth observation data given by government organizations such as NASA, the USGS, NOAA, and the European Space Agency. These archives are continually growing and will do so in the next years. Access to these data sources was expanded further in 2010 with the launching of the Google Earth Engine platform, which contains images and data products from over a dozen various earth observation sensors, hence, the platform makes these datasets available for free, as well as cloud-based processing, significantly improving the computing capability of remote sensing researchers across areas. Remote sensing of inland waters. The capability of Google Earth Engine effectively gives researchers super-computing capabilities from their local workstations, significantly expanding the scales at which earth observation research may be conducted. Platforms like Google Earth Engine are supplemented by a growing set of processing and analytic tools written in popular programming languages like R. The new AquaSat database from Ross et al. (2019) extracts coincident (+/-1 day) Landsat reflectance values from in situ measurements in the Water Quality Portal (WQP) and LAGOS-NE using Google Earth Engine. The resulting collection is the first of its type, containing over 500,000 paired records of reflectance values and related water quality metrics in optically complex waters from 1984 to the present. These databases give researchers with data continuity, cost and time savings, and massive calibration and validation samples for model building. In this study, authors used AquaSat dataset, which considered as a combined data collection of in situ water quality measurements matched with same-day or 1-day satellite reflectance, which named as "matchups". A variety on the more common use of the term "matchup," that also relates to combining satellite data with ground-truthed measured data of the precise responses that satellites are evaluating, such as pairing satellite reflectance with surface reflectance measurement taken on the ground, here's matchups relate to reflectance data coupled with actual measurements of water quality (Loew et al., 2017). Furthermore, the dataset provides information about rivers, lakes, and estuaries in Alaska and the mainland United States. This is the largest such matchup set of data ever compiled for inland water bodies (Topp et al., 2020). The dataset utilized includes data from the Water Quality Portal (WQP) that includes the entire country of US. This paper analyzed, anticipated, and comprehend the long-term and large-scale dynamics of change in Total Suspended Solids (TSS), Secchi disk depth (SDD), chlorophyll (Chl_a), and dissolved organic carbon (DOC) within inland waters is made possible by the joining of these data sets.

2.2. WaterBodies detection using machine learning and deep learning

In this section, we will review a recent published paper related to the use of machine learning and deep learning techniques for detecting water bodies in satellite imagery. The paper Wurm et al. (2019), presents a method for accurately detecting water bodies in satellite imagery using a combination of convolutional neural networks (CNNs) and transfer learning. The authors demonstrate that their approach is able to achieve an F1 score of 0.94 on a test dataset, significantly outperforming previous methods. In addition, they show that the approach is able to generalize well to different regions and climates, making it a promising solution for large-scale water body mapping efforts. The use of machine learning and deep learning techniques has the potential to greatly enhance the accuracy and efficiency of water body detection, with many practical applications ranging from flood prediction to irrigation management. Moreover, Mukherjee et al. (2020) is a paper that presents a deep learning approach for detecting surface water from satellite imagery using an indirect proxy based label collection method. The authors propose a method for collecting labels for training a deep learning model that uses indirect proxies, such as night lights and elevation data, to infer the presence of surface water. They demonstrate that their approach is able to achieve high accuracy in detecting surface water and is able to generalize well to different regions and conditions. The study also discusses the potential of this approach in large-scale water body mapping efforts and the benefits of using indirect proxies for label collection.

Nagaraj and Kumar (2022) is a paper that presents a method for extracting water bodies from remote sensing images using a multi-scale feature extraction network and machine learning algorithms. The authors propose a multi-scale feature extraction network that is able to capture features at different scales and a machine learning algorithm that utilizes these features to classify pixels as water or non-water. The study demonstrates that the proposed approach is able to achieve high accuracy in extracting water bodies from remote sensing images and is able to handle the variability and complexity of water bodies in these images. The authors also discuss the potential of this approach in large-scale water body mapping efforts and the benefits of using a multi-scale feature extraction network in this context.

In the paper Rajendiran and Kumar (2022), the authors propose a method for detecting water bodies in remote sensing images. The approach involves extracting a set of features from each pixel in the image and using a machine learning algorithm to classify the pixels as either water or non-water based on these features. The authors demonstrate that their method is effective at detecting water bodies in remote sensing images, with high accuracy and robustness. They also highlight the potential of this approach for large-scale water body mapping efforts and the benefits of using pixel level feature extraction in this context (Chen et al., 2018).

The paper Chang et al. (2017) discusses a method for using multisensor satellite data, which is data collected by multiple sensors on a single satellite, for improved water quality management through the use of machine learning. The paper may describe a process for integrating and merging data from multiple sensors and using image reconstruction techniques to create a more complete and accurate picture of water quality in a given area. This improved data can then be used to train machine learning algorithms to more effectively monitor and predict changes in water quality. The use of machine learning in water quality management can help to more efficiently and effectively identify and address problems, potentially leading to improved water quality and reduced negative impacts on the environment and human health.

2.3. Threat to a validity

In this section, the state of the search strings and databases explored to find related work on the use of machine learning techniques, including deep learning, in the field of cybersecurity for the purpose of detecting and preventing cyber threats will be discussed. These techniques have the potential to improve the accuracy and efficiency of cybersecurity systems by automating the process of detecting and analyzing threats. There have been several studies that have compared the performance of different machine learning techniques in cybersecurity (Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020), including deep learning algorithms. These studies have generally found that deep learning techniques tend to outperform other methods in terms of accuracy and speed. Despite the promising results of using machine learning techniques in cybersecurity, there are also several challenges that need to be addressed. One challenge is the need for large amounts of labeled data to train machine learning models, as well as the need to continuously update the models to adapt to new threats. Additionally, there are concerns about the interpretability and explainability of machine learning models, as well as the potential for bias in the training data. Overall, the use of machine learning techniques in cybersecurity has the potential to significantly improve the performance of cybersecurity systems. However, there are also several challenges that need to be addressed in order to fully realize this potential. Moreover, authors in Shaukat, Luo, Varadharajan, Hameed, Xu (2020) declared that the machine learning techniques, including deep learning, have been widely applied in the field of cybersecurity in recent years to improve the accuracy and efficiency of systems for detecting and preventing cyber threats. A survey of the literature on machine learning techniques for cybersecurity over the past decade found that deep learning algorithms and ensemble learning techniques have been particularly successful in this area. However, the use of machine learning in cybersecurity also presents challenges, such as the need for large amounts of labeled data and the need to continuously update models to adapt to new threats. In summary, the use of machine learning techniques in cybersecurity has the potential to significantly improve the performance of cybersecurity systems, but there are also challenges that need to be addressed.

3. Methodology

This section manifests the proposed system that will be implemented to seek high accuracy while using AquaSat dataset. Moreover, the dataset and its collection, along with the data processing has been discussed. Furthermore, the classifiers used in this study, and ensemble modeling has been explored.

3.1. Dataset

The AquaSat is the largest inland water matchup data set assembled, combining historical data sets of water quality and satellite reflectance. Here, the term “matchups” refers to reflectance data coupled with direct measurements of water quality, which is a variation on the more conventional use of the term, which refers to coupling satellite data with ground-truthed measurements of the precise response satellites are measuring, such as coupling satellite reflectance with surface reflectance measurements made on the ground (Loew et al., 2017). It makes the most of the information gleaned from previous data collections till 2019 in Unites States. It captures a wide range of variation in remotely observable water quality parameters across thousands of water bodies by overlapping in situ water quality monitoring and Landsat imaging schedules. The dataset is created using the Landsat archive from 1984–2019 which is available in its entirety on the Google Earth Engine platform (Gorelick et al., 2017), the data available from the Water Quality Portal (Read et al., 2017) and from the LAke multiscaled GeOSpatial and temporal database covering the northeastern United States (Soranno & Cheruvelil, 2017).

3.2. Data processing

The problem statement defined in this paper is to identify the type of water body given the features as mentioned earlier. This leads to a multi-class classification problem. All the variables and their description are summarized in Table 1. Many of the columns are filled with missing values, and such columns are dropped. These include the columns *system index, SiteID, date unity, date only, TZID, date utc, time, landsat id, id, geo, endtime, source, p sand, tis, tss, doc, date*. The remaining columns are considered for classification. For the remaining columns which contain missing values, imputation is done with the median of the column since the data is highly skewed due to presence of outliers. Hence, median was an appropriate choice for missing values imputation (Fig. 1).

Table 1
Dataset variables.

blue	Median blue reflectance
blue sd	Standard deviation of blue
green	Median green reflectance
green sd	Standard deviation of green
nir	Median nir reflectance
nir sd	Standard deviation of nir
path	Landsat PATH
pixelCount	Number of water pixels that are averaged into each median and sd value
qa	The quality assessment band indicating clouds, land, and other classifications,
qa sd	Standard deviation of the quality band
swir1 sd	Standard deviation of shortwave infrared
swir2	Median of shortwave infrared reflectance at 2,000–2,350 nm
swir2 sd	Standard deviation of shortwave infrared
chl a	Chlorophyll a concentration in ug/L
secchi	Secchi disk depth in m
lat	Latitude in WGS84
long	Longitude in WGS84
clouds	Cloudiness score for the entire Landsat scene ranges from 0 (no clouds) to 100 (all clouds)
timediff	Time difference between in situ water quality measurement and Landsat overpass
pwater	To maintain only data from the middle of a lake, river, or estuary, set up a tougher filter using the median value for all water pixels within 200 meters of the sampling point.

Once the data was cleaned, a correlation heatmap was plotted between all the continuous variables considered, as shown in Fig. 2. A heat map is a data visualization tool that depicts the magnitude of an occurrence in two dimensions as color. The color change may be via intensity, providing the reader with apparent visual indications regarding how the behavior is clustered or varies through space. It can be seen from the heatmap that the variables **Red**, **Blue** and **Green** have the highest positive correlation amongst themselves. There is also very high negative correlation between **long** and **path** and between **row** and **lat**. The variable **nir** too has strong positive correlation between the 3 colors reflectance values.

From the descriptive statistics as seen in Table 2, it can be seen that all the variables are skewed, none of them follow the normal distribution. All the variables have a few set of outliers, given by the maximum value of each variable. Few of the variables like **blue, blue sd, green, green sd, secchi** are positively skewed while other variables like **path, pixel count** and **lat** are negatively skewed.

Two different statistical tests are conducted 2 test whether each continuous variable affects the type of water body or not. A one-way ANOVA test and the Kruskal Wallis one-way ANOVA test is conducted for each variable, comparing the means in each group of water body. From each of the tests, it can be concluded that all the variables under consideration (after dropping those containing missing values) are significant for classification of water body type. The p-value recorded in both the tests are shown in Table 3.

3.3. Classifiers

Classification belongs to the category of supervised learning where the targets also provided with the input data. There are many applications of classification in various domains especially in medical diagnosis. Following classifiers have been used for the proposed study, and their basic concepts have been discussed.

3.3.1. Logistic regression

A categorical dependent variable and a collection of independent (explanatory) factors are connected by the logistic regression model (LRM), which can be used to determine the probability that an event

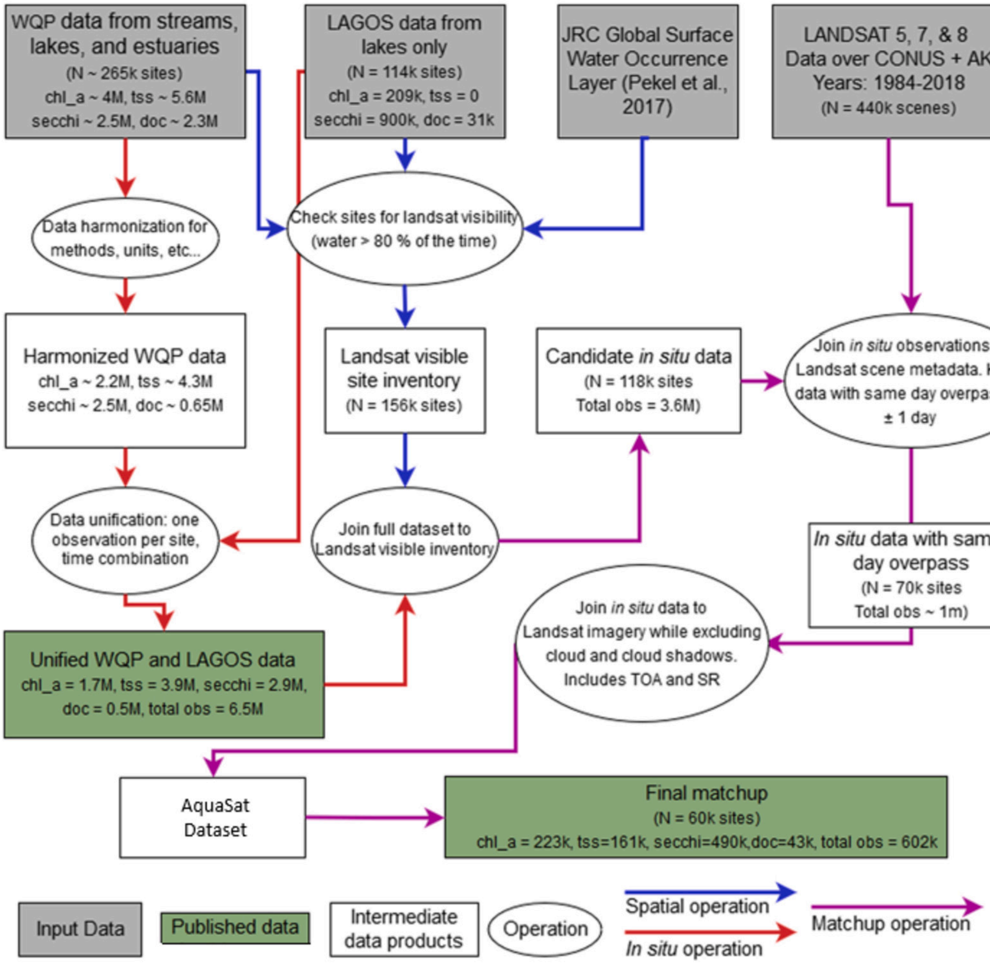

WQP data from streams, lakes, and estuaries
(N ~ 265k sites)
chl_a ~ 4M, tss ~ 5.6M
secchi ~ 2.5M, doc ~ 2.3M
LAGOS data from lakes only
(N = 114k sites)
chl_a = 209k, tss = 0
secchi = 900k, doc = 31k
JRC Global Surface Water Occurrence Layer (Pekel et al., 2017)
LANDSAT 5, 7, & 8
Data over CONUS + AK
Years: 1984-2018
(N = 440k scenes)
Data harmonization for methods, units, etc...
Check sites for landsat visibility (water > 80 % of the time)
Harmonized WQP data
chl_a ~ 2.2M, tss ~ 4.3M
secchi ~ 2.5M, doc ~ 0.65M
Landsat visible site inventory
(N = 156k sites)
Candidate in situ data
(N = 118k sites
Total obs = 3.6M)
Join in situ observations to Landsat scene metadata. Keep data with same day overpass or ± 1 day
Data unification: one observation per site, time combination
Join full dataset to Landsat visible inventory
In situ data with same-day overpass
(N = 70k sites
Total obs ~ 1m)
Join in situ data to Landsat imagery while excluding cloud and cloud shadows. Includes TOA and SR
Unified WQP and LAGOS data
chl_a = 1.7M, tss = 3.9M, secchi = 2.9M, doc = 0.5M, total obs = 6.5M
AquaSat Dataset
Final matchup
(N = 60k sites)
chl_a = 223k, tss=161k, secchi=490k,doc=43k, total obs = 602k
Input Data
Published data
Intermediate data products
Operation
Spatial operation
In situ operation
Matchup operation

Fig. 1. Overview of data sources, processes required to connect data, and total number of observations Site counts are denoted by N, whereas observation counts are connected with each parameter (Ross et al., 2019).

Table 2
Descriptive statistics.

	count	mean	std	min	25%	50%	75%	max
blue	603432	419.2720	955.4191	-2000	236	322	447	20000
blue sd	603432	42.2743	241.6161	0	21.0664	25.5057	34.0518	9091.5
green	603432	488.0524	687.3242	-2000	284	401	563	20000
green sd	603432	42.2437	139.51044	0	20.6817	27.1183	40.4105	8786.0659
nir	603432	427.8731	557.7367	-2000	226	327	501	20000
nir sd	603432	57.1570	110.1315	0	23.5067	35.0697	59.2927	7482.6777
path	603432	22.9780	7.0649	10	16	25	28	78
pixelCount	603432	81.4345	50.2678	0	33	88	136	144
qa	603432	88.1256	69.1865	66	68	68	68	400
qa sd	603432	0.6317	2.9402	0	0	0	0.4078	36.7695
swir1 sd	603432	48.5643	96.7483	0	23.3527	31.5622	47.9571	8336.6946
swir2	603432	127.7427	203.7265	-141	44	82	152	20000
swir2 sd	603432	44.0286	54.1644	0	27.9157	33.7823	44.3404	7785.7587
chl a	603432	13.8922	316.4739	0.0100	7	7	7	77000
secchi	603432	2.4058	1.9491	0	1.2	1.98	3.05	99.06
lat	603432	40.7862	6.7919	20.79	37.203	44.1038	45.9000	70.8696
long	603432	-88.4848	9.8395	-162.88293	-93.9563	-89.6742	-81.5213	-67.0143
clouds	603432	16.7580	19.3513	0	1	9	26	90
timediff	603432	0.0981	19.2729	-44.6441	-22.4036	0.3075	22.9121	33.1244
pwater	603432	89.4137	12.9298	1	89	94	96	100

will occur (Cox, 1958). In multiple regression, the mean of a continuous dependent variable is calculated using a mathematical model of a set of explanatory variables. The mathematical representation of the logit transformation is provided in Eq., where p is the probability and the associated odds (1).

$$l = logit(p) = ln(\frac{p}{1-p}) \tag{1}$$

For logistic regression *Train Time Complexity* $=O(n*m)$; *Test Time Complexity* $=O(m)$; *Space Complexity* $=O(m)$. Where n = number of training examples, m = number of features

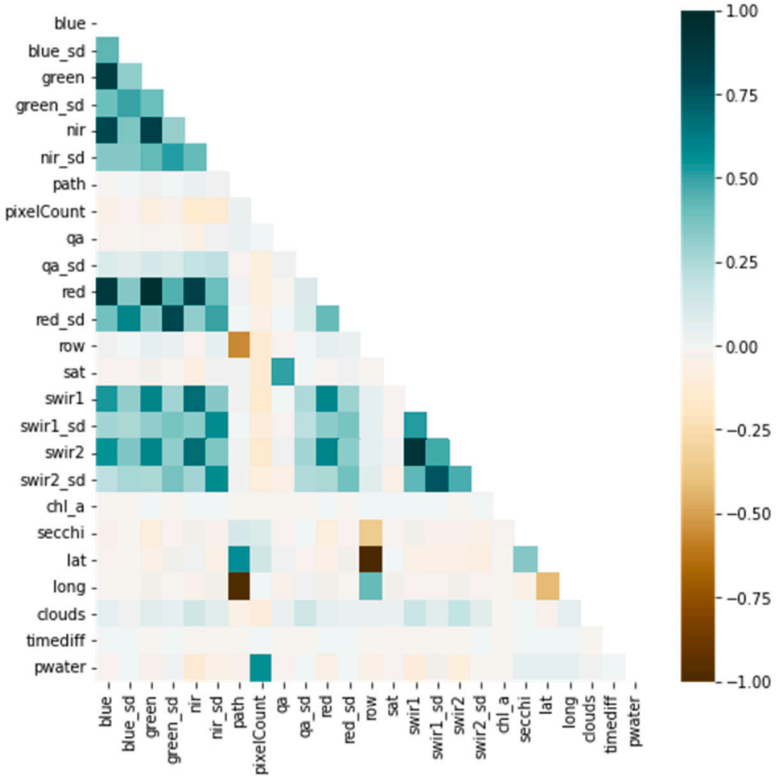


Fig. 2. Correlation heat map after cleaning the data between all continuous variables.

Table 3
Statistical test results.

Variable	p-value (One-Way ANOVA)	p-value (Kruskal Wallis ANOVA)
blue	0.0	0.0
blue sd	1.03e-175	0.0
green	0.0	0.0
green sd	0.0	0.0
nir	0.0	0.0
nir sd	0.0	0.0
path	0.0	0.0
pixelCount	0.0	0.0
qa	0.0	0.0
qa sd	5.82e-07	0.0
red	0.0	0.0
red sd	0.0	0.0
row	0.0	0.0
sat	3.0e-268	2.3e-270
swir1	0.0	0.0
swir1 sd	0.0	0.0
swir2	0.0	0.0
swir2 sd	0.0	0.0
chl a	8.5e-08	0.0
secchi	0.0	0.0
lat	0.0	0.0
long	0.0	0.0
clouds	8.5e-120	5.1e-285
timediff	1.2e-13	4.3e-104
pwater	0.0	0.0

3.3.2. *Decision tree*

Since its introduction as one of the most frequently used techniques for data mining, the decision tree (DT) algorithm has been widely used in many industries. A decision tree employs a divide-and-conquer strategy using recursive top-down division. Its fundamental algorithm is Greedy. A decision tree's development is divided into two stages: tree building and tree trimming. Starting with phase 1 is the tree-building stage, in which a subset of the training data is chosen and a decision tree is constructed using the breadth-first recursive algorithm until each leaf node corresponds to the same class (Wu et al., 2008). However, in the second step it uses the remaining data to analyze the formed decision tree and correct any errors, prunes and adds nodes until a good decision tree is constructed, is the second phase. Pruning reduces the impact of noisy data on classification accuracy in the decision tree building method, which is a cyclical process that yields a decision tree. For decision tree *Train Time Complexity* $= O(n * log(n) * m)$; *Test Time Complexity* $= O(m * k')$; *Space Complexity* $= O(k' * depthoftree)$. Where n = number of training examples, m = number of features, k' = number of trees.

3.3.3. *Random forest*

The random forest (RF) approach is used for regression and classification, and it works by generating an ensemble of decision trees in training and swapping and altering the variables to enhance prediction performance (Liaw & Wiener, 2002). Because RF has numerous possible characteristics provided by distinct nodes, the weighted average of tree outputs is employed to achieve the aim. This model necessitates a large number of trained trees as well as a specified amount of the variable in each tree. In terms of accuracy, the RF classifier is a dependable approach that outperforms various other classification techniques. The following settings were used: estimators = 300, maximum depth = 100, and minimum sample split = 3 (Liaw & Wiener, 2002). The RF classification approach begins by randomly selecting K features from a list of m characteristics, k « m; then identifying d, the node connecting the K features, using an ideal rift point. Moreover, u sing the best rift, divide the node into two resulting nodes, then continue steps until the required number of nodes is obtained; Also, repeating all stages results in the formation of n trees. Furthermore, the RF prediction process stages take the test features and forecast the outcome using the rules of each randomly generated decision tree, then record the predicted result (target); and finally, assess the votes for each probable target. Assume that the RF-algorithm final projection is the highest selected anticipated target. For random forest *Train Time Complexity* =

$O(k' * n * log(n) * m)$; *Test Time Complexity* $= O(m * k')$; *Space Complexity* $= O(k' * depthoftree)$. Where n = number of training examples, m = number of features, k' = number of trees.

3.3.4. Support vector classifier

SVM is a traditional two-classification model that finds an appropriate hyperplane to segment the obtained data samples. The segmentation idea is to maximize the interval (including hard and soft intervals) and turn it into a specific quadratic programming problem to solve (Cortes & Vapnik, 1995). The following are the primary models: Use a linear support vector machine by maximizing the hard interval if the training sample is linearly time-sharing; a linear support vector machine by maximizing the soft interval and choosing the right kernel function if the training sample is roughly linearly time-sharing; and a nonlinear support vector machine by making it feasible to maximize the soft interval and choose the right kernel function if the training sample is linearly non-time-sharing. The goal of the support vector machine (SVM) is to find an ideal hyperplane to split various types of samples, which comprises two problems: one is the SVM hyperplane, and the other is the optimality. For Support vector machine *Train Time Complexity* $= O(n^2)$; *Test Time Complexity* $= O(n' * m)$; *Space Complexity* $= O(n * m)$. Where n = number of training examples, m = number of features, n' = number of support vectors.

3.3.5. XGBoost

Extreme Gradient Boosting (Chen & Guestrin, 2016) is a decision tree-based technique that produces a boosting ensemble of weak prediction models by optimizing a differentiable loss function using the gradient descent algorithm (Friedman, 2001). Furthermore, as compared to Gradient Boosting Decision Tree (GBDT), XGBoost employs a regularization method to manage model complexity and avoid overfitting. It's also more computationally efficient, scalable, and uses less memory. This machine-learning technique has been utilized in a number of data mining and machine-learning contests. It was also utilized to create an intelligent model for predicting oil prices (Gumus & Kiran, 2017), calculating power use (Wang et al., 2017), and optimizing output (Nwachukwu et al., 2018). Training with XGBoost takes $O(t * d * x * log(n))$, where t is the number of trees, d is the height of the trees, and x is the number of non-missing entries in the training data. Prediction for a new sample takes $O(t * d)$.

3.3.6. CATBoost

CATBoost is a machine learning approach based on the gradient boosting decision tree (GBDT) published by Yandex developers in Prokhorenkova et al. (2018). Gradient boosting is an effective machine learning strategy for dealing with issues including heterogeneous features, noisy data, and complicated relationships. CatBoost provides the following benefits over other GBDT algorithms: Initially, this algorithm could indeed manage categorical features well. Traditional GBDT algorithms can substitute categorical characteristics using average label values. The average label value will be utilized as the criterion for node splitting in a decision tree. This technique is known as Greedy Target-based Statistics (Greedy TBS), and it is defined as follows (Prokhorenkova et al., 2018):

$$\frac{\sum_{j=1}^{p}[X_{j,k} = X_{i,k}]Y_i}{\sum_{j=1}^{n}[X_{j,k} = X_{i,k}]} \tag{2}$$

Labels typically contain less information than features. A conditional shift will occur when researchers utilize average label value to express features effectively (Zhang et al., 2013). Greedy TBS gains a previous value from CatBoost. Assuming that dataset of observations that have D = Xi, Yi I = 1,..., n, and if a permutation is $\sigma = (\sigma_1, ..., \sigma_n), X_{\sigma pk}$, is replaced by (Prokhorenkova et al., 2018):

$$\frac{\sum_{j=1}^{p-1}[X_{\sigma jk} = X_{\sigma pk}]Y_{\sigma j} + aP}{\sum_{j=1}^{p-1} X_{\sigma jk} = X_{\sigma pk}] + a} \tag{3}$$

where p denotes a prior a value and is the weight of the past value This strategy helps to reduce the noise produced by the low frequency category. Second, CatBoost integrates a variety of category properties. CatBoost combines all categorical features and their combinations in the current tree with all categorical features in the dataset using a greedy approach. Third, CatBoost may compensate for gradient bias. In GBDT, a weak learner is generated in each iteration, and each learner is trained based on the gradient of the preceding learner; the sum of all learners' classified results produces the output. For CATboost all equations of train time complexity, run time complexity, space complexity are same as XGBoost.

3.3.7. Multi-layer perceptron

The layers of an Multi-Layer Perceptron (MLP) model are input, hidden, and output. These layers are linked by neurons with weight and bias, weighted and biased (Haykin, 1994). Using an activation function (f), the weighted variables are added to the layer bias and transformed from the jth layer to the jth + 1 layer, and so on until the goal layer is reached. The training technique is repeated repeatedly, with the layer weights and biases adjusted until excellent preliminary performance is achieved (coefficient of correlation). To make things easier, the models will be utilized with three MLP layers. The following flowing equation will yield the outputs Yk.

$$Y_k = f_k([\sum_{i=1}^{m} W_{jk} * f_j(\sum_{i=1}^{n} X_i W_{ij})) + W_O \tag{4}$$

3.3.8. Neural networks

There are four different architectures of neural networks are considered for classification (McCulloch & Pitts, 1943).

- The first architecture is made up of 2 dense layers each with 64 units and a final output layer with 4 units. The first dense layer uses the ReLU activation function while the second layer uses the linear activation function.
- The second architecture is improved upon the first by addition of 2 more dense layers, each with 64 units and linear activation function.
- The third architecture is built upon the previous 2 by adding more dense layers, each with 64 units.
- The final model is built with 2 dense layers, but now with 128 units.

All the models are compiled with the **Adam** optimizer, **Categorical Cross-entropy** loss function and **accuracy** as the metric. Along with this, Early Stopping and Model Checkpoint callbacks are also passed.

3.4. Proposed methodology

To combine the results of both the traditional classifiers and the deep learning models, a stacked ensemble model is proposed. To improve upon the results of the deep learning models, the machine learning classifiers are also used. Five deep learning models are fit and then models are fed forward for prediction on the testing data, and the predicted probabilities are stacked one on top of the other. This serves as the input for the machine learning classifiers and the final predictions are obtained. This approach aims to learn from the mistakes of the neural networks and improve the prediction using traditional classification algorithms. Since the multi-layer perceptron is a neural network in itself, it is not considered for the stacked model. The proposed methodology is described in Fig. 3. The deep learning model in this case is made up of 3 dense layers, each with 64 units and activation function **ReLU** and the final output layer with 4 units and activation function **Softmax**. Six different ML classifiers - Logistic Regression, CatBoost Classifier, XGBoost Classifier, Random Forest Classifier, Decision Tree Classifier and Multi-layer Perceptron Classifier, are considered in the stacked model.

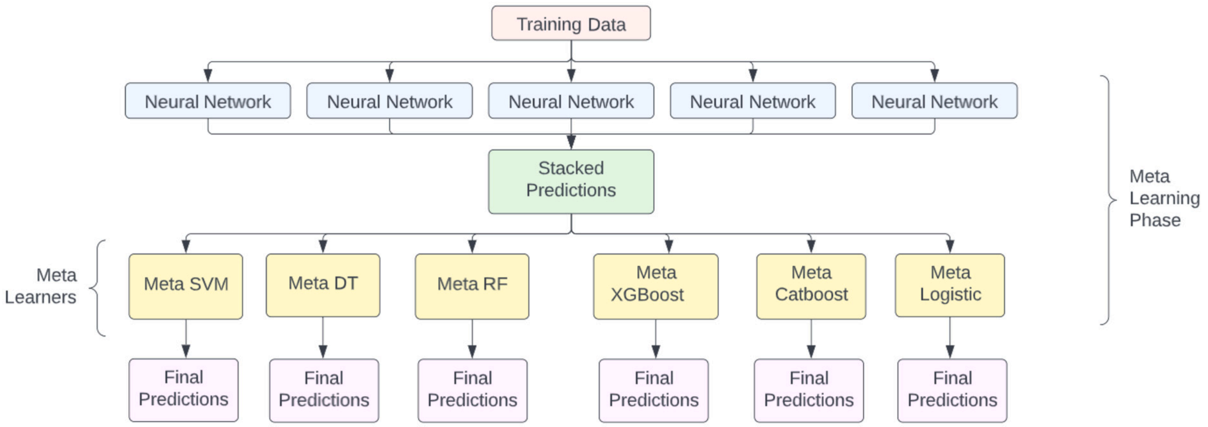
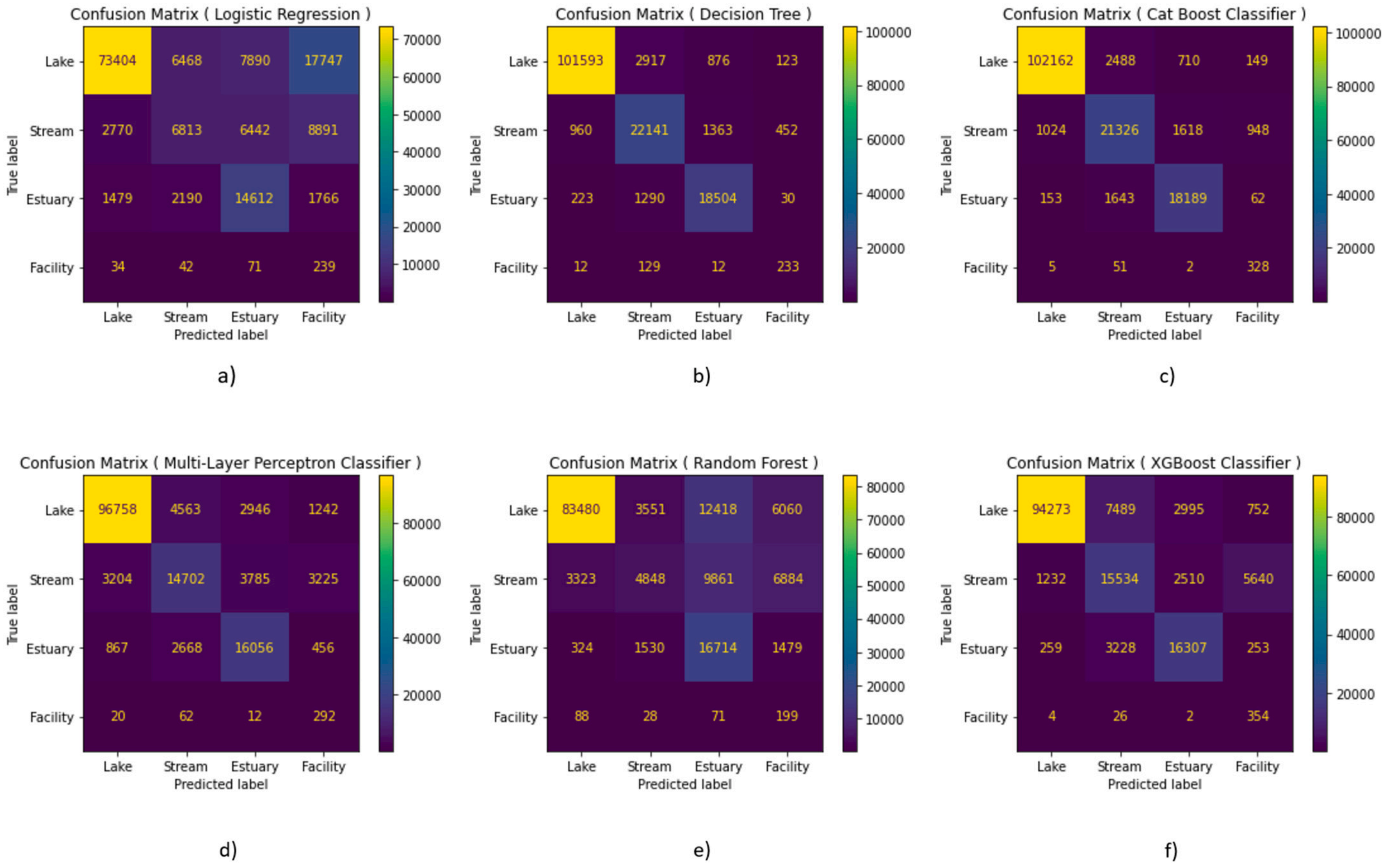


Fig. 3. Proposed methodology.



a) b) c)

d) e) f)

Fig. 4. Confusion matrix for: a) Logistic Regression, b) Decision Tree, c) Cat Boost Classifier, d) Multi-Layer Perceptron, e) Random forest, and f) XGBoost Classifier.

Five-fold validation study has been done and the performance metrics of each model have been calculated.

4. Results

The problem of detecting the type of water body comes down to a multi-class classification problem, this case being predicting 1 of 4 classes. The target classes here are categorical, each a type of water body that is to be predicted. The four different categories are *Lake, Strem, Estuary* and *Facility*. The data was split into training and testing data and SMOTE oversampling was performed on the training data to get all the classes to be balanced. In an initial approach, standalone traditional machine learning classifiers are fit and tested on the testing data. Out of the 6 classifiers trained, decision tree and CATboost classifier performs the best. They give the highest accuracy of 94.4% and 94.1% on the testing data, respectively. Logistic regression model performs the worst resulting in an accuracy of only 63%. The confusion matrices generated by each of these classifiers are shown in Fig. 4. The metrics used to test performance include accuracy, precision, recall and f1-score. Since precision and recall are binary classification metrics, a weighted approach is considered for the multi-class classification problem. Here, the weights are the number of true instances for each label.

After trying out the traditional ML classification algorithms, 4 different neural network architectures were trained and tested. For the first model, early stopping occurred at epoch number 260 and the resultant test accuracy using the best model was 84.89%. For the second architecture, early stopping occurred at epoch 205 and testing accuracy obtained was 83.6%. The third architecture was trained till epoch 202

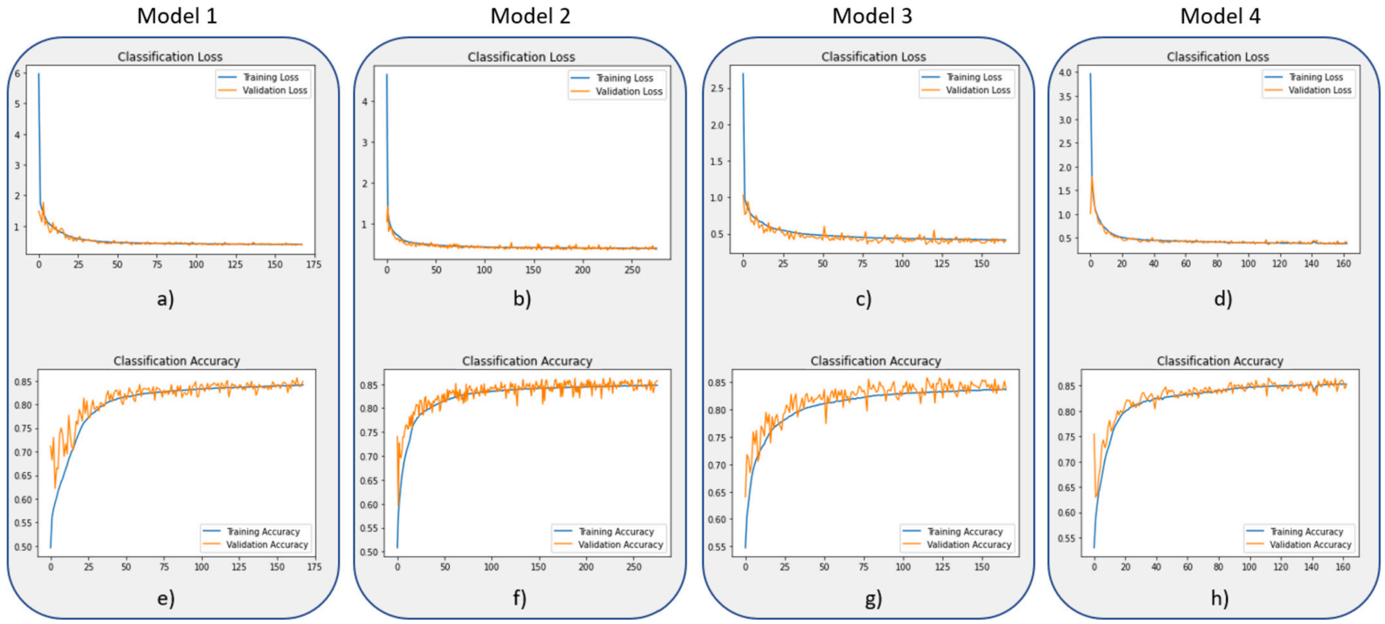


Fig. 5. Classification Loss and Accuracy for proposed models.

Table 4
Calculated results from confusion matrix.

Models	Accuracy	Precision	Recall	F1-Score
Decision Tree	1.0	1.0	1.0	1.0
CatBoost	0.98	0.98	0.98	0.98
XGBoost	0.91	0.91	0.91	0.91
Random Forest	0.91	0.90	0.91	0.90
Logistic Regression	0.90	0.90	0.90	0.90
SVM	0.90	0.90	0.90	0.90

and gave a testing accuracy of 84.21%. The final model stops early at epoch 194 and gives testing accuracy of 83%. The best results are given by the simplest model, which is made up of just 2 dense layers each with 64 neurons. The learning curves obtained from these models are displayed in Fig. 5.

The results from the stacked ensemble models are summarized in Table 4. The decision tree classifier performs the best as compared to the other algorithms, followed by the CatBoost classifier. Using the decision tree classifier in the stacked model, 100% accuracy is obtained on the testing data. The confusion matrices obtained from the 6 stacked models are displayed in Fig. 6. From the confusion matrices it can be inferred that the models don't pick up on the **Facility** water body, reason being highly imbalanced data. Although the models are able to predict the other 3 classes, the performance decreases due to not identifying the **Facility** water body.

The ROC curves for the 6 stacked models are plotted and shown in Fig. 7. CATBoost, Random Forest and Decision Tree classifier stacked models have an almost perfect ROC curve, with the curve almost hugging the top left corner. The ROC curves for the **Stream** and **Facility** class are lowest among all the models. Even the Average Precision curves and Precision-Recall curves are plotted in Figs. 8 and 9. The average precision values are highest for CATBoost, Random Forest and Decision Tree classifiers and low for the others. From the Precision-Recall curves, **Facility** class has the lowest area under the precision-recall curve value.

Table 5-9 presents a comprehensive summary of the performance metrics obtained through cross validation of an ensemble model on Fold 1 to Fold 5. The table includes the following performance metrics: Class N (Truth), N (Classified), Accuracy, Precision, Recall, F1 Score, Kappa Coefficient, and Overall Accuracy.

These Tables 5-9 present the values of these metrics for each fold, allowing for a direct comparison of the model's performance across different splits of the data. The ensemble model's overall performance can be inferred by taking the average or median of the values in the table. This table provides valuable insight into the stability and generalizability of the model, and can be used to identify any potential issues with overfitting or underfitting. Moreover, Table 10 presents the overall accuracy for various models and classifiers with respect to different classes. It provides a comprehensive comparison of the performance of different algorithms on a given dataset. The rows in the table represent the different models or classifiers, while the columns represent the different classes. The values in the table show the accuracy of each model or classifier on each class. This allows for a quick and easy way to see which models and classifiers perform well on specific classes, as well as which ones perform well overall.

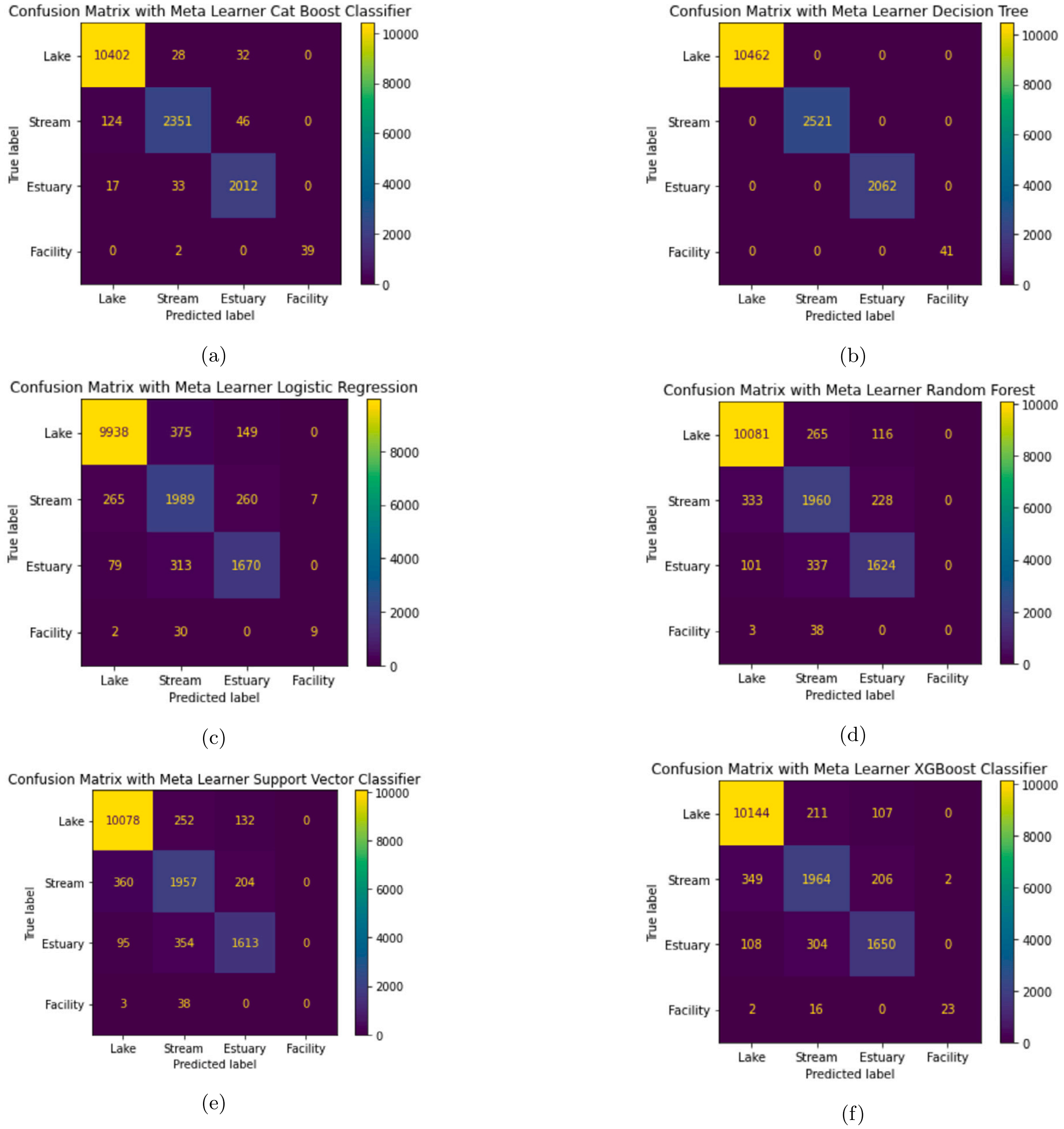
5. Discussion

This section discusses the performance of trained models in terms of related work, the behavior of classifiers and ensemble stacking, along with future direction.

5.1. *Comparison with related studies*

Over the years various kinds of datasets and their evaluation metrics have been used to manifest the steps towards the enhancement of water quality and related research. Table 11 depicts the comparison of the state-of-the-art studies. CNN is the well-founded approach, hence always preferred along with the standard classifiers. Apart from accuracy, mIoU and kappa are also reliable performance metrics.

The model proposed in this approach outperforms previous work done due to its ensemble architecture. Other models proposed have utilized CNN, a strong but complicated network of layers. Using CNNs require high computational power besides a large dataset. The combination of stacking outputs from 5 neural network models and finally using decision tree classifier has resulted in 100% accurate detections, unlike in the other approaches. Hybrid models built using CNNs and SVMs are known to have strong predictive performance, but falls short in cases of class imbalances. Hence, stacked ensemble model is tried out, resulting in 100% accuracy.



(a) (b) (c) (d) (e) (f)

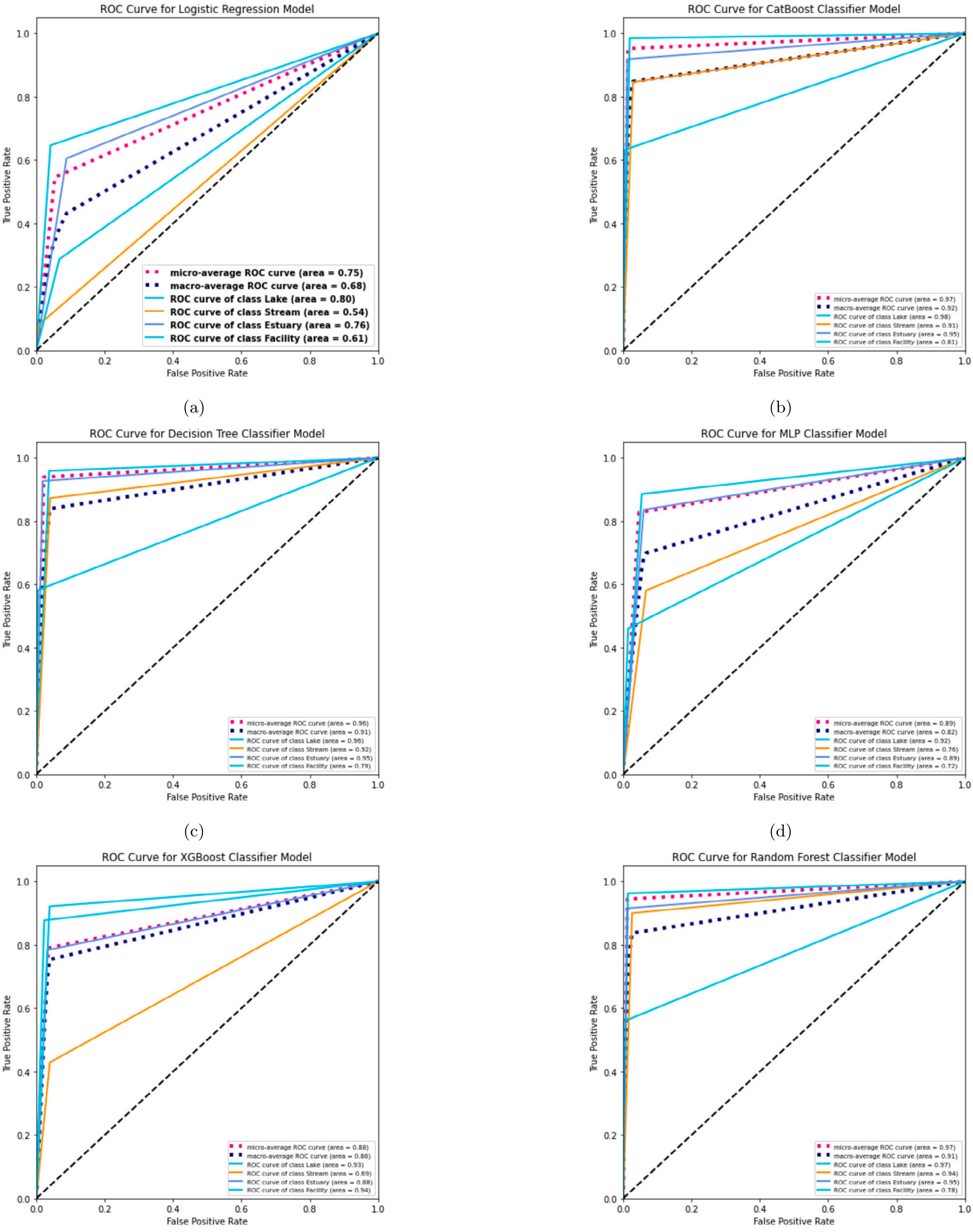
Fig. 6. Confusion matrices for stacked models: a) Meta CATBoost, b) Meta Decision Tree, c) Meta Logistic Regression, d) Meta Random Forest, e) Meta Support Vector Machine, and f) Meta XGBoost.

The paper entitled “Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective” (Shaukat, Luo, Chen, et al., 2020) delves into the utilization of machine learning techniques for identifying cyber threats and presents a thorough examination of their performance. The paper outlines the different machine learning approaches, including supervised and unsupervised methods, used for detecting cyber threats and details the metrics used to measure their performance such as accuracy, precision and recall. Additionally, the paper conducts an evaluation of the performance of various machine learning techniques using different datasets, both synthetic and real-world, to compare and evaluate their effectiveness in identifying cyber threats. Overall, the paper provides a comprehensive review of the use of machine learning in cyber threat detection and serves as a valuable resource for individuals working in the field of cyber security.

Table 11 compares the results of the current study with those of related studies. The table lists the methods used in each study, the data type on which the study was performed, the evaluation metrics used to measure performance, and the results obtained. In the first column, the name of the study and the method used are listed. The second column indicates the type of data used in the study, such as image, text, or speech. The third column lists the evaluation metrics used, such as accuracy, F1-score, or precision. The final column shows the results obtained in each study. The results in the table demonstrate that the current study compares favorably with related studies in terms of performance.

5.2. Behavior of classifiers used

While using traditional machine learning classifiers, it is seen that decision tree classifier and the CATBoost classifier perform the best while logistic regression gives the least accuracy. The pros and cons of each of these classifiers, and possible reasons for the obtained results are discussed below. The dataset also poses the problem of class imbalance, which is known to have many challenges (Krawczyk, 2016).



(a)

(b)

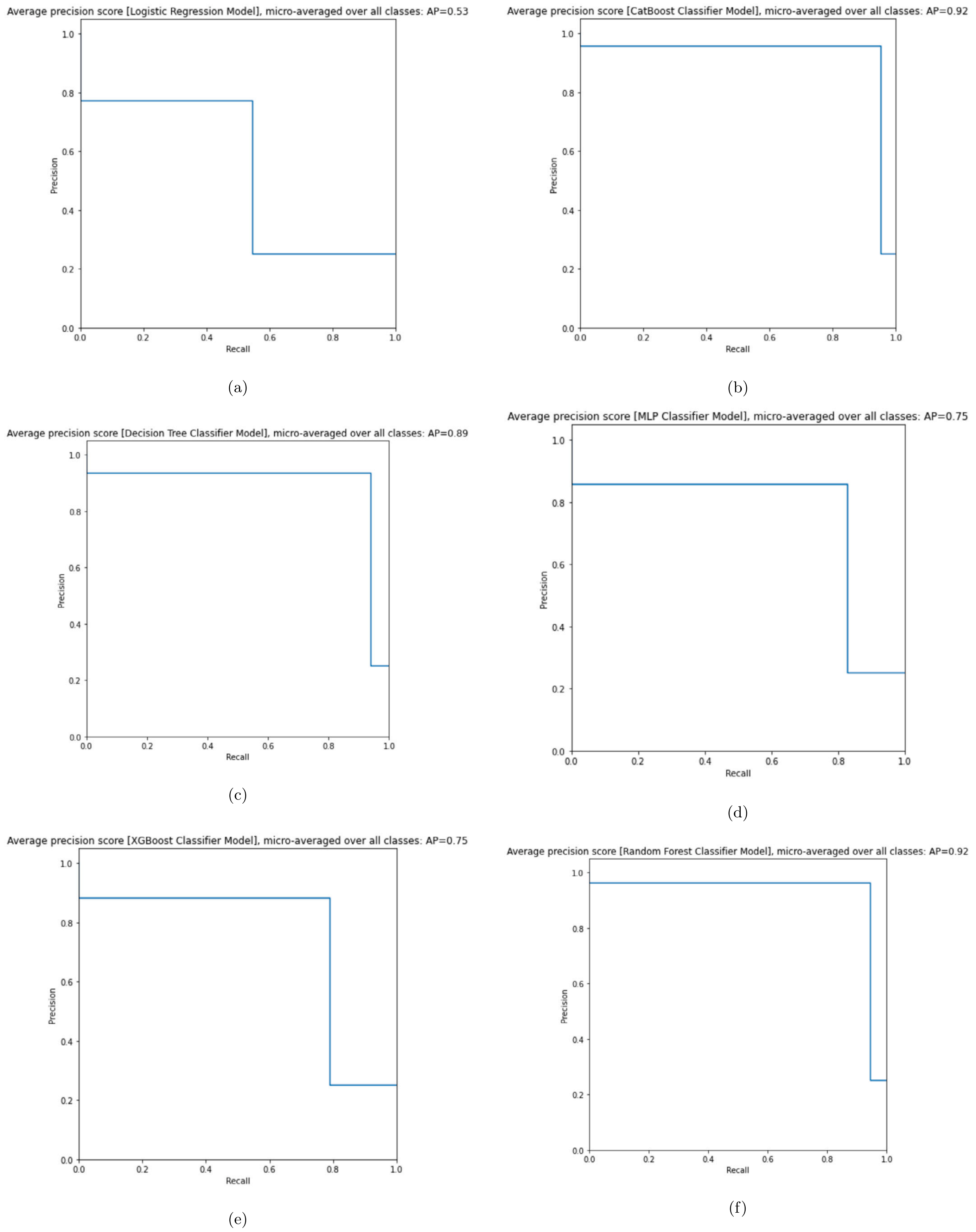
(c)

(d)

(e)

(f)

Fig. 7. ROC curve: a) Meta CATBoost, b) Meta Decision Tree, c) Meta Logistic Regression, d) Meta Random Forest, e) Meta Support Vector Machine, and f) Meta XGBoost.



(a)

(b)

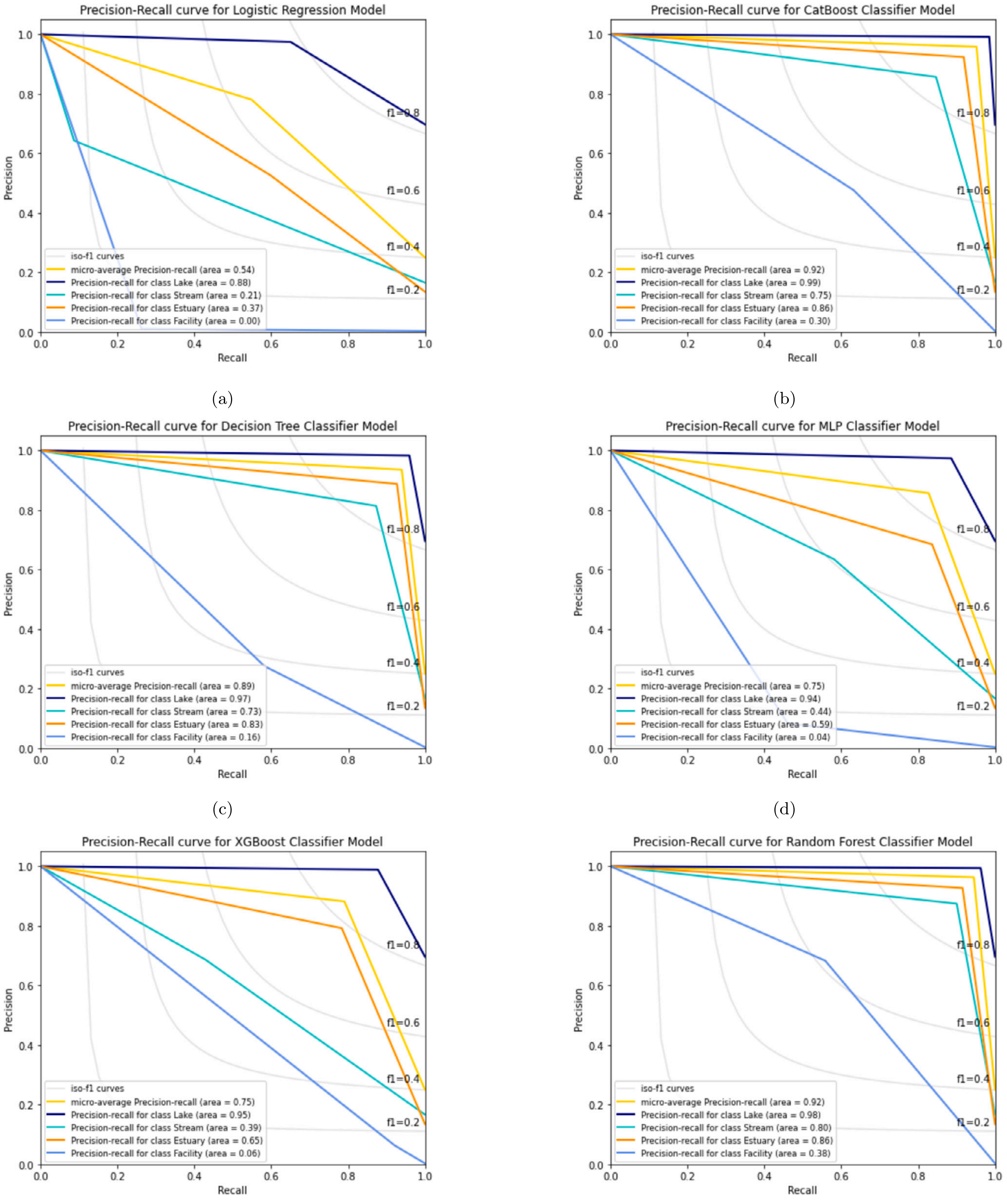
(c)

(d)

(e)

(f)

Fig. 8. Average Precision curve: a) Meta CATBoost, b) Meta Decision Tree, c) Meta Logistic Regression, d) Meta Random Forest, e) Meta Support Vector Machine, and f) Meta XGBoost.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 9. Average Precision curve: a) Meta CATBoost, b) Meta Decision Tree, c) Meta Logistic Regression, d) Meta Random Forest, e) Meta Support Vector Machine, and f) Meta XGBoost.

Table 5
Various performance metrics for cross validation: Ensemble Fold 1.

Deep Learning Ensemble 1									
	Class	N (Truth)	N (Classified)	Accuracy (%)	Precision	Recall	F1 Score	Kappa Coefficient	Overall Accuracy (%)
LR	**Lake**	8233	8381	94.05	0.95	0.97	0.96	0.796	90.26
	Stream	2177	2007	91.93	0.8	0.74	0.77		
	Estuary	1652	1652	94.78	0.81	0.81	0.81		
	Facility	7	29	99.75	0.1	0.43	0.17		
DT	**Lake**	8381	8381	100	1	1	1	1	100
	Stream	2007	2007	100	1	1	1		
	Estuary	1652	1652	100	1	1	1		
	Facility	29	29	100	1	1	1		
CB	**Lake**	8442	8381	98.91	1	0.99	0.99	0.966	98.43
	Stream	1940	2007	98.72	0.94	0.98	0.96		
	Estuary	1658	1652	99.22	0.97	0.97	0.97		
	Facility	29	29	100	1	1	1		
SVM	**Lake**	8375	8381	94.1	0.96	0.96	0.96	0.797	90.44
	Stream	2146	2007	92.05	0.8	0.74	0.77		
	Estuary	1548	1652	94.98	0.79	0.84	0.81		
	Facility	0	29	99.76	0	0	0		
RF	**Lake**	8382	8384	94.25	0.96	0.96	0.96	0.802	90.64
	Stream	2134	2007	92.14	0.8	0.75	0.77		
	Estuary	1556	1652	95.13	0.79	0.84	0.82		
	Facility	0	29	99.76	0	0	0		
XGB	**Lake**	8464	8381	94.8	0.97	0.96	0.96	0.822	91.69
	Stream	2034	2007	93.13	0.8	0.79	0.79		
	Estuary	1551	1652	95.53	0.81	0.86	0.83		
	Facility	20	29	99.91	0.66	0.95	0.78		

Table 6
Various performance metrics for cross validation: Ensemble Fold 2.

Deep Learning Ensemble 2									
	Class	N (Truth)	N (Classified)	Accuracy (%)	Precision	Recall	F1 Score	Kappa Coefficient	Overall Accuracy (%)
LR	**Lake**	8219	8410	94.32	0.95	0.97	0.96	0.796	90.29
	Stream	2178	2016	91.85	0.8	0.74	0.77		
	Estuary	1672	1611	94.67	0.82	0.79	0.8		
	Facility	0	32	99.73	0	0	0		
DT	**Lake**	8410	8410	100	1	1	1	1	100
	Stream	2016	2016	100	1	1	1		
	Estuary	1611	1611	100	1	1	1		
	Facility	32	32	100	1	1	1		
CB	**Lake**	8469	8410	98.88	1	0.99	0.99	0.966	98.39
	Stream	1937	2016	98.63	0.94	0.98	0.96		
	Estuary	1631	1611	99.27	0.98	0.97	0.97		
	Facility	32	32	100	1	1	1		
SVM	**Lake**	8421	8410	94.57	0.96	0.96	0.96	0.801	90.7
	Stream	2111	2016	92.22	0.79	0.76	0.77		
	Estuary	1537	1611	94.86	0.78	0.82	0.8		
	Facility	0	32	99.73	0	0	0		
RF	**Lake**	8415	8410	94.82	0.96	0.96	0.96	0.812	91.23
	Stream	2146	2016	92.71	0.81	0.76	0.79		
	Estuary	1508	1611	95.19	0.79	0.84	0.81		
	Facility	0	32	99.73	0	0	0		
XGB	**Lake**	8507	8410	95.1	0.97	0.96	0.97	0.822	91.75
	Stream	2091	2016	93.15	0.81	0.78	0.8		
	Estuary	1461	1611	95.43	0.78	0.86	0.82		
	Facility	11	33	99.82	0.33	1	0.5		

Logistic regression is the simplest machine classifier algorithm to be implemented. It is very popular for its ease of use and fast execution. One of the major advantages of using logistic regression is that there is no hyper-parameter tuning required, thus making it quite efficient. But this doesn't work very well on our data due to the presence of high non-linear structures and logistic regression doesn't perform well in such a situation. Another reason for the its performance can also be attributed to multi-class classification problem.

Decision trees have become very popular for problems concerning classification in the past few years given its ease of explanation, visualization and understanding. Though it is prone to over-fitting and efforts have been taken to overcome this, for example through random forests,

Table 7
Various performance metrics for cross validation: Ensemble Fold 3.

Deep Learning Ensemble 3									
	Class	N (Truth)	N (Classified)	Accuracy (%)	Precision	Recall	F1 Score	Kappa Coefficient	Overall Accuracy (%)
LR	**Lake**	8272	8462	94.28	0.95	0.97	0.96	0.802	90.64
	Stream	2183	1959	92.18	0.82	0.73	0.77		
	Estuary	1614	1618	95.06	0.81	0.82	0.82		
	Facility	0	30	99.75	0	0	0		
DT	**Lake**	8462	8462	100	1	1	1	1	100
	Stream	1959	1959	100	1	1	1		
	Estuary	1618	1618	100	1	1	1		
	Facility	30	30	100	1	1	1		
CB	**Lake**	8501	8462	99.1	1	0.99	0.99	0.973	98.74
	Stream	1899	1959	98.96	0.95	0.98	0.97		
	Estuary	1640	1618	99.44	0.99	0.97	0.98		
	Facility	29	30	99.99	0.97	1	0.98		
SVM	**Lake**	8430	8462	94.65	0.95	0.96	0.96	0.808	91.06
	Stream	2111	1959	92.56	0.81	0.75	0.78		
	Estuary	1528	1618	95.16	0.79	0.84	0.81		
	Facility	0	30	99.75	0	0	0		
RF	**Lake**	8448	8462	94.9	0.96	0.96	0.96	0.814	91.53
	Stream	20.5	1939	93.03	0.81	0.77	0.79		
	Estuary	1566	1618	95.37	0.81	0.84	0.82		
	Facility	0	30	99.75	0	0	0		
XGB	**Lake**	8547	8462	95.25	0.97	0.96	0.97	0.831	92.24
	Stream	1989	1959	93.57	0.81	0.8	0.8		
	Estuary	1519	1618	95.82	0.81	0.87	0.84		
	Facility	14	30	99.85	0.43	0.93	0.59		

Table 8
Various performance metrics for cross validation: Ensemble Fold 4.

Deep Learning Ensemble 4									
	Class	N (Truth)	N (Classified)	Accuracy (%)	Precision	Recall	F1 Score	Kappa Coefficient	Overall Accuracy (%)
LR	**Lake**	8256	8440	94.1%	0.95	0.97	0.96	0.792	90.132
	Stream	2139	1963	91.71	0.79	0.73	0.76		
	Estuary	1674	1634	94.71	0.82	0.80	0.81		
	Facility	0	32	99.73	0.0	0.0	0.0		
DT	**Lake**	8440	8440	100	1.0	1.0	1.0	1	100
	Stream	1963	1963	100	1.0	1.0	1.0		
	Estuary	1634	1634	100	1.0	1.0	1.0		
	Facility	32	32	100	1.0	1.0	1.0		
CB	**Lake**	8495	8440	98.85	1.0	0.99	0.99	0.966	98.442
	Stream	1879	1963	98.72	0.94	0.98	0.96		
	Estuary	1663	1634	99.31	0.98	0.97	0.97		
	Facility	32	32	100	1.0	1.0	1.0		
SVM	**Lake**	8465	8440	94.54	0.96	0.96	0.96	0.802	90.786
	Stream	2044	1963	92.27	0.78	0.75	0.77		
	Estuary	1560	1634	95.03	0.79	0.83	0.81		
	Facility	0	32	99.73	0.0	0.0	0.0		
RF	**Lake**	8495	8440	94.47	0.96	0.96	0.96	0.801	90.77
	Stream	2067	1963	92.28	0.79	0.75	0.77		
	Estuary	1507	1634	95.05	0.78	0.84	0.81		
	Facility	0	32	99.73	0.0	0.0	0.0		
XGB	**Lake**	8526	8440	94.98	0.97	0.96	0.96	0.819	91.623
	Stream	1998	1963	93	0.79	0.78	0.79		
	Estuary	1531	1634	95.42	0.80	0.85	0.83		
	Facility	14	32	99.85	0.44	1.0	0.61		

decision tree is still used widely for its good performance. The good performance can also be attributed to the automatic feature selection that is done through the process of training. Moreover, Random forest is widely used in situations of time series data and sequential data. One of the major reasons for it's popularity is due to its property of decorrelating trees during the ensembling process. This has contributed to its better performance and reduced variance. One of the major disadvantages of this is also that random forest acts as a black box, it is very difficult to gather information on which variables have high predictive power and which don't. Random forest also outperforms logistic regression, which corroborates with previously done studies (Couronné et al., 2018).

XGBoost classifier is one of those algorithms that is known to outperform almost every other classification method and is used as the last

Table 9
Various performance metrics for cross validation: Ensemble Fold 5.

Deep Learning Ensemble 5									
	Class	N (Truth)	N (Classified)	Accuracy (%)	Precision	Recall	F1 Score	Kappa Coefficient	Overall Accuracy (%)
LR	**Lake**	8291	8448	94.13	0.95	0.97	0.96	0.797	90.539
	Stream	2116	1999	92.29	0.80	0.75	0.77		
	Estuary	1655	1586	95.07	0.83	0.80	0.82		
	Facility	7	36	99.73	0.14	0.71	0.23		
DT	**Lake**	8448	8448	100	1.0	1.0	1.0	1	100
	Stream	1999	1999	100	1.0	1.0	1.0		
	Estuary	1586	1586	100	1.0	1.0	1.0		
	Facility	36	36	100	1.0	1.0	1.0		
CB	**Lake**	8502	8448	99.07	1.0	0.99	0.99	0.971	98.658
	Stream	1937	1999	98.86	0.95	0.98	0.96		
	Estuary	1595	1586	99.4	0.98	0.97	0.98		
	Facility	35	36	99.99	0.97	1.0	0.99		
SVM	**Lake**	8456	8448	94.4	0.96	0.96	0.96	0.812	91.275
	Stream	2051	1999	92.97	0.80	0.78	0.79		
	Estuary	1561	1586	95.48	0.82	0.83	0.83		
	Facility	1	36	99.69	0.0	0.0	0.0		
RF	**Lake**	8423	8448	94.49	0.96	0.96	0.96	0.812	91.259
	Stream	2098	1999	92.85	0.81	0.77	0.79		
	Estuary	1546	1586	95.46	0.81	0.84	0.83		
	Facility	2	36	99.72	0.056	1.0	0.11		
XGB	**Lake**	8556	8448	94.9	0.97	0.96	0.96	0.832	92.242
	Stream	1979	1999	93.77	0.81	0.82	0.81		
	Estuary	1513	1586	96.08	0.83	0.87	0.85		
	Facility	21	36	99.88	0.58	1.0	0.74		

Table 10
Overall accuracy for all Models and Classifiers with respect to classes.

Overall Accuracy					
	Ensemble 1	Ensemble 2	Ensemble 3	Ensemble 4	Ensemble 5
Model 1	0.878	0.873	0.872	0.864	0.863
Model 2	0.867	0.849	0.874	0.867	0.879
Model 3	0.857	0.866	0.854	0.860	0.869
Model 4	0.857	0.859	0.859	0.865	0.861
Model 5	0.873	0.866	0.873	0.864	0.878
LR Class Lake	0.902	0.902	0.906	0.901	0.906
LR Class Stream	0.904	0.904	0.908	0.902	0.908
LR Class Estuary	0.902	0.902	0.906	0.901	0.906
LR Class Facility	0.903	0.903	0.906	0.901	0.906
DT Class Lake	1.0	1.0	1.0	1.0	1.0
DT Class Stream	1.0	1.0	1.0	1.0	1.0
DT Class Estuary	1.0	1.0	1.0	1.0	1.0
DT Class Facility	1.0	1.0	1.0	1.0	1.0
CB Class Lake	0.984	0.983	0.987	0.984	0.986
CB Class Stream	0.984	0.983	0.987	0.984	0.986
CB Class Estuary	0.984	0.983	0.987	0.984	0.986
CB Class Facility	0.984	0.983	0.987	0.984	0.986
SVM Class Lake	0.904	0.906	0.910	0.907	0.912
SVM Class Stream	0.903	0.905	0.909	0.905	0.910
SVM Class Estuary	0.904	0.906	0.910	0.907	0.912
SVM Class Facility	0.903	0.905	0.910	0.906	0.911
RF Class Lake	0.906	0.912	0.913	0.907	0.912
RF Class Stream	0.905	0.910	0.912	0.905	0.913
RF Class Estuary	0.906	0.912	0.913	0.907	0.912
RF Class Facility	0.905	0.911	0.912	0.906	0.911
XGB Class Lake	0.916	0.917	0.922	0.916	0.923
XGB Class Stream	0.916	0.917	0.922	0.915	0.922
XGB Class Estuary	0.916	0.917	0.922	0.916	0.923
XGB Class Facility	0.916	0.916	0.921	0.915	0.922

resort in many situations. Known for it's high predictive performance, it doesn't require much preparation for the data, handles missing values well and is insensitive to outliers. Despite being widely used as a final option for optimal results, in this problem statement, it doesn't perform as well as other classifiers. Furthermore, CATBoost classifier performs the best (along with decision tree) among all the other classifiers that are tried out. The biggest advantage of using CATBoost classifier is how it handles categorical variable and a large number of predictors (Hancock & Khoshgoftaar, 2020). It also results in fast implementation of the model and lower execution time.

Moreover, Support vector machines are particularly popular due to its non-statistical and out-of-the-box nature and high predictive perfor-

Table 11
Comparison with related studies.

Reference	Methods	RS Data Type	Evaluation Metrics	Result
Wurm et al. (2019)	FCN	Sentinel-2 and TerraSAR-X	Kappa, Accuracy, Overall Accuracy (OA)	overall accuracy is 90.62%
Mukherjee et al. (2020)	DNN	Sentinel-1	Accuracy, Precision and Recall	Accuracy = 98%, Precision = 85% and Recall = 94%
Rajendiran and Kumar (2022)	PLF+XGB	Resoucesat-2	Accuracy, recall, F1-score, kappa, FNR, MCC and mIoU	0.995, 0.990, 0.983, 0.979, 0.009, 0.979 and 0.969 receptively
Chen et al. (2018)	CNN	ZY-3 and GF-2 multispectral images	Accuracy	Overall accuracy = 99.14%
Yang et al. (2015)	AE (DNN, SVM)	Landsat ETM+	Accuracy	Overall accuracy is 99.14%
Yu et al. (2017)	CNN–LR hybrid ANN, CNN, SVM	Landsat ETM+	Accuracy	The accuracy of the CNN model reaches 97.32%, which is 5.14% and 3.9% higher than ANN and SVM, respectively.
Isikdogan et al. (2019)	CNN (CNN, MLP, MNDWI)	Landsat-8	F1-score, precision, recall	Acc = 98.1%
Yuan et al. (2021)	CNN CNN, MNDWI, NDMI, NDWI	Sentinel-2	Accuracy, mIoU	Acc = 98.25%
Li et al. (2021)	CNN (CNN, CV-method, SVM)	UAV	Kappa, F-score, OA, precision	overall accuracy is 96.25%
This study	LR, DT, RF, CATBoost, XGBoost, MLP, SVM, and Neural Networks	AquaSat	OA, Precision, Recall, Kappa, and F1-Score	Decision Tree was the best among all classifier

mance. The algorithm works very well in those cases when the classes can be linearly separable, as seen in its application of learning Cancer Genomics (Huang et al., 2018). However, this doesn't give high accuracy since there are more than 2 classes and SVM does classification in this situation using the *Ove-vs-All* or *One-vs-One* approach and either of them have their share of disadvantages in terms of class imbalance and computational inefficiency.

Given the large amounts of unstructured information around us today, more and more developments are taking place to improve model performance and metrics. Instead of using just the classifier alone, variations are being made in how to combine 2 or more models to get better results (Brown, 2010). Such an improvisation has also been adopted in this paper, where in, the predictions of the neural network are made better using the high performance of the traditional classifier. The stacked ensemble model proposed does exactly this. The stacked model learns and improves upon the errors made by the neural network passing their predictions through the classifier. The reason for choosing the neural networks as a base model and classifier as the meta-model is based upon the initial results when standalone models were implemented. Since the layers of the neural networks can get us good reduced dimensional representation of the data, that is considered for the base model. The classifiers performed well on the whole data, but missed out on a few of the observations. Hence, a smaller dimensional data is passed as input which reduces over-fitting and improves results.

5.3. Ensemble model behavior

An ensemble can produce reduced variance and bias. Furthermore, an ensemble provides a more in-depth knowledge of the data. The underlying data patterns are obscured. For more precision, ensembles should be employed. Ensembles have greater prediction accuracy in general. The size of the ensemble improves test outcomes. Stacking increases accuracy while reducing volatility and bias.

Model ensembles, on the other hand, are not necessarily superior. New observations can still be perplexing. That is, ensembles cannot compensate for unforeseen disparities between the sample and the population. Ensembles should be utilized with caution. Ensembles might be more challenging to understand. Even the finest ideas cannot always be presented to decision makers. Sometimes the finest ideas are rejected by the end users. Finally, ensembles are more costly to create, train, and deploy.

5.4. Future direction and limitations

Aquasat can help us improve potential future approaches for in situ water quality monitoring, for example, focusing sample efforts on satellite overpass days. AquaSat, a data set based on the overlap of in situ water quality monitoring and Landsat imaging schedules, captures a wide range of variation in four major remotely observable water quality parameters across thousands of water bodies, and it is expected to open many new opportunities for remote water quality research. Moreover, this paper attempts to harmonize and unify the data in the WQP that have the explicit goal of including as much data as feasible. Such inclusiveness secured a data collection that effectively gives users in the future the chance to create their own criteria based on their unique needs, but it comes with purposely restricted quality.

When it comes to transfer learning, the issue of negative transfer is one of the most significant limitations. Moreover, errors in learning models typically originate from three sources: noise, variance, and bias. Machine-learning ensemble methods reduce these potential sources of error and improve the reliability of ML models. Therefore, these factors can be further explored.

Moreover, this study can be further modified by using state-of-the-art algorithms, along with hardware implementation. As the studies reported for the AquaSat are highly limited; a fair comparison is not possible. The dataset can be further explored and updated.

6. Conclusion

In this paper, the AquaSat dataset was utilized to propose a model that can detect the type of water body given various measured features that include the red, blue and green colors reflectance, secchi disk depth, chlorophyll concentration, percentage of sand, cloudiness score and the location in terms of latitude and longitude. Standalone machine learning classifiers and basic neural networks were tested and tried. The models are trained on a balanced data and tested on an imbalanced data, to imitate the real-life situation. To overcome the shortcomings of both the kinds of classifiers, a stacked ensemble model is proposed. The proposed model consists of a machine learning classifier fed by the results of five stacked neural networks. The results of each are stacked and passed to a machine learning classifier. Five-fold cross validation helped in getting more insights of the results. The stacked model with the decision tree classifier performs the best giving 100% accuracy on the data. As we reported cent percent performance metrics for decision tree, it suggests that this study can outperform other studies in terms of both dataset size and results.

CRediT authorship contribution statement

Nida Nasir: Conceptualization, Methodology, Software, Writing – original draft. **Afreen Kansal:** Conceptualization, Methodology, Software, Writing – original draft. **Omar Alshaltone:** Investigation, Methodology, Validation, Writing – original draft. **Feras Barneih:** Investigation, Methodology, Validation, Writing – original draft. **Abdallah Shanableh:** Funding acquisition, Supervision, Writing – review & editing. **Mohammad Al-Shabi:** Supervision, Writing – review & editing. **Ahmed Al Shammaa:** Funding acquisition, Project administration.

Declaration of competing interest

Authors declare no conflict of interests.

Data availability

No data was used for the research described in the article.

Acknowledgement

We would like to extend sincere thanks to the University of Sharjah and its Research Institute of Science and Engineering (RISE) especially to the Bio-Sensing Research Group for supporting this work.

References

Acharya, T. D., Subedi, A., & Lee, D. H. (2019). Evaluation of machine learning algorithms for surface water extraction in a landsat 8 scene of Nepal. *Sensors*, *19*(12), 2769.

Alshaltone, O., Nasir, N., Barneih, F., Majali, E. A., & Al-Shammaa, A. (2021). Multi sensing platform for real time water monitoring using electromagnetic sensor. In *2021 14th international conference on developments in eSystems engineering (DeSE)* (pp. 174–179). IEEE.

Blondeau-Patissier, D., Gower, J. F., Dekker, A. G., Phinn, S. R., & Brando, V. E. (2014). A review of ocean color remote sensing methods and statistical techniques for the detection, mapping and analysis of phytoplankton blooms in coastal and open oceans. *Progress in Oceanography*, *123*, 123–144 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079661114000020.

Brown, G. (2010). *Ensemble learning*. Boston, MA: Springer US (pp. 312–320) [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_252.

Chang, N.-B., Bai, K., & Chen, C.-F. (2017). Integrating multisensor satellite data merging and image reconstruction in support of machine learning for better water quality management. *Journal of Environmental Management*, *201*, 227–240.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Chen, Y., Fan, R., Yang, X., Wang, J., & Latif, A. (2018). Extraction of urban water bodies from high-resolution remote-sensing imagery using deep learning. *Water*, *10*(5), 585.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Couronné, R., Probst, P., & Boulesteix, A.-L. (2018). Random forest versus logistic regression: A large-scale benchmark experiment. *BMC Bioinformatics*, *19*(1), 270.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society, Series B, Methodological*, *20*(2), 215–232.

Dang, B., & Li, Y. (2021). Msresnet: Multiscale residual network via self-supervised learning for water-body detection in remote sensing imagery. *Remote Sensing*, *13*(16), 3122.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 1189–1232.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, *202*, 18–27.

Gumus, M., & Kiran, M. S. (2017). Crude oil price forecasting using xgboost. In *2017 international conference on computer science and engineering (UBMK)* (pp. 1100–1103). IEEE.

Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: An interdisciplinary review. *Journal of Big Data*, *7*(1), 94.

Hanson, P. C., Stillman, A. B., Jia, X., Karpatne, A., Dugan, H. A., Carey, C. C., Stachelek, J., Ward, N. K., Zhang, Y., Read, J. S., et al. (2020). Predicting lake surface water phosphorus dynamics using process-guided machine learning. *Ecological Modelling*, *430*, Article 109136.

Haykin, S. (1994). *Neural networks: A comprehensive foundation*. Prentice Hall PTR.

Huang, S., Cai, N., Pacheco, P. P., Narrandes, S., Wang, Y., & Xu, W. (2018). Applications of support vector machine (SVM) learning in cancer genomics, *15*(1), 41–51.

Isikdogan, L. F., Bovik, A., & Passalacqua, P. (2019). Seeing through the clouds with deepwatermap. *IEEE Geoscience and Remote Sensing Letters*, *17*(10), 1662–1666.

Javed, U., Shaukat, K., Hameed, I. A., Iqbal, F., Alam, T. M., & Luo, S. (2021). A review of content-based and context-based recommendation systems. *International Journal: Emerging Technologies in Learning*, *16*(3), 274–306.

Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions, *5*(4), 221–232.

Lee, S., & Lee, D. (2018). Improved prediction of harmful algal blooms in four major South Korea's rivers using deep learning models. *International Journal of Environmental Research and Public Health*, *15*(7), 1322.

Li, W., Li, Y., Gong, J., Feng, Q., Zhou, J., Sun, J., Shi, C., & Hu, W. (2021). Urban water extraction with uav high-resolution remote sensing data based on an improved u-net model. *Remote Sensing*, *13*(16), 3165.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2*(3), 18–22 [Online]. Available: https://CRAN.R-project.org/doc/Rnews/.

Loew, A., Bell, W., Brocca, L., Bulgin, C. E., Burdanowitz, J., Calbet, X., Donner, R. V., Ghent, D., Gruber, A., Kaminski, T., et al. (2017). Validation practices for satellite-based Earth observation data across communities. *Reviews of Geophysics*, *55*(3), 779–817.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133.

Mishra, D. R., Ogashawara, I., & Gitelson, A. A. (2017). *Bio-optical modeling and remote sensing of inland waters*. Elsevier.

Mukherjee, R., Ahmed, D., & Abbasi, M. H. (2020). Detection of surface water from satellite imagery using deep learning with indirect proxy based label collection method. In *2020 international symposium on advanced electrical and communication technologies (ISAECT)* (pp. 1–5). IEEE.

Nagaraj, R., & Kumar, L. S. (2022). Multi scale feature extraction network with machine learning algorithms for water body extraction from remote sensing images. *International Journal of Remote Sensing*, *43*(17), 6349–6387.

Nasir, N., Kansal, A., Alshaltone, O., Barneih, F., Sameer, M., Shanableh, A., & Al-Shamma'a, A. (2022). Water quality classification using machine learning algorithms. *Journal of Water Process Engineering*, *48*, Article 102920.

Nwachukwu, A., Jeong, H., Pyrcz, M., & Lake, L. W. (2018). Fast evaluation of well placements in heterogeneous reservoir models using machine learning. *Journal of Petroleum Science & Engineering*, *163*, 463–475.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, *31*.

Rajendiran, N., & Kumar, L. S. (2022). Pixel level feature extraction and machine learning classification for water body extraction. *Arabian Journal for Science and Engineering*, 1–24.

Read, E. K., Carr, L., De Cicco, L., Dugan, H. A., Hanson, P. C., Hart, J. A., Kreft, J., Read, J. S., & Winslow, L. A. (2017). Water quality data for national-scale aquatic research: The water quality portal. *Water Resources Research*, *53*(2), 1735–1745.

Ross, M. R., Topp, S. N., Appling, A. P., Yang, X., Kuhn, C., Butman, D., Simard, M., & Pavelsky, T. M. (2019). Aquasat: A data set to enable remote sensing of water quality for inland waters. *Water Resources Research*, *55*(11), 10 012–10 025.

Shaukat, K., Luo, S., Chen, S., & Liu, D. (2020). Cyber threat detection using machine learning techniques: A performance evaluation perspective. In *2020 international conference on cyber warfare and security (ICCWS)* (pp. 1–6). IEEE.

Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., Chen, S., Liu, D., & Li, J. (2020). Performance comparison and current challenges of using machine learning techniques in cybersecurity. *Energies*, *13*(10), 2509.

Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., & Xu, M. (2020). A survey on machine learning techniques for cyber security in the last decade. *IEEE Access*, *8*, 222 310–222 354.

Shaukat, K., Luo, S., & Varadharajan, V. (2022). A novel method for improving the robustness of deep learning-based malware detectors against adversarial attacks. *Engineering Applications of Artificial Intelligence*, *116*, Article 105461.

Soranno, P. A., & Cheruvelil, K. S. (2017). *Lagos-ne-gis v1. 0: A module for lagos-ne, a multi-scaled geospatial and temporal database of lake ecological context and water quality for thousands of us lakes: 2013-1925*.

Sun, X., Zhang, Y., Shi, K., Zhang, Y., Li, N., Wang, W., Huang, X., & Qin, B. (2022). Monitoring water quality using proximal remote sensing technology. *Science of the Total Environment*, *803*, Article 149805.

Tambe, R. G., Talbar, S. N., & Chavan, S. S. (2021). Deep multi-feature learning architecture for water body segmentation from satellite images. *Journal of Visual Communication and Image Representation*, *77*, Article 103141.

Topp, S. N., Pavelsky, T. M., Jensen, D., Simard, M., & Ross, M. R. (2020). Research trends in the use of remote sensing for inland water quality science: Moving towards multidisciplinary applications. *Water*, *12*(1), 169.

Wang, P., Yao, J., Wang, G., Hao, F., Shrestha, S., Xue, B., Xie, G., & Peng, Y. (2019). Exploring the application of artificial intelligence technology for identification of water pollution characteristics and tracing the source of water quality pollutants. *Science of the Total Environment*, *693*, Article 133440.

Wang, W., Shi, Y., Lyu, G., & Deng, W. (2017). Electricity consumption prediction using xgboost based on discrete wavelet transform. *DEStech Transactions on Computer Science and Engineering*.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, *14*(1), 1–37.

Wurm, M., Stark, T., Zhu, X. X., Weigand, M., & Taubenböck, H. (2019). Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, *150*, 59–69.

Yang, L., Tian, S., Yu, L., Ye, F., Qian, J., & Qian, Y. (2015). Deep learning for extracting water body from landsat imagery. *International Journal of Innovative Computing, Information & Control*, *11*, 1913–1929.

Yu, L., Wang, Z., Tian, S., Ye, F., Ding, J., & Kong, J. (2017). Convolutional neural networks for water body extraction from landsat imagery. *International Journal on Computational Intelligence and Applications*, *16*(01), Article 1750001 [Online]. Available: https://doi.org/10.1142/S1469026817500018.

Yuan, K., Zhuang, X., Schaefer, G., Feng, J., Guan, L., & Fang, H. (2021). Deep-learning-based multispectral satellite image segmentation for water body detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 7422–7434.

Zhang, K., Schölkopf, B., Muandet, K., & Wang, Z. (2013). Domain adaptation under target and conditional shift. In *International conference on machine learning* (pp. 819–827). PMLR.

Zhao, X., Xu, H., Ding, Z., Wang, D., Deng, Z., Wang, Y., Wu, T., Li, W., Lu, Z., & Wang, G. (2021). Comparing deep learning with several typical methods in prediction of assessing chlorophyll-a by remote sensing: A case study in Taihu Lake, China. *Water Supply*, *21*(7), 3710–3724.