Water Potability Prediction and Comparation Using Decision Tree, SVM and ANN

Zahra Fayyadiyati
5025201133
zahrafayyadiyati.ac@gmail.com

Wina Tungmiharja
5025201242
winatungmiharja@gmail.com

Nazhifah Elqolby
5025201156
line 3:
nelqolby@gmail.com

Luthfiyyah Hanifah Amari
5025201090
luthfiyyah.hanifah@gmail.com

Water quality could determine the potability of a water body. The characteristics between potable water and unpotable water is hard to discern by people who used to live around a water body as they have used to consume it on daily basis. Therefore, it would be convenient if a predictor based on artificial intelligence for water potability is created. Water quality indicators from 3276 different water sources are included in our dataset to train and test the model based on 9 properties. There will be three computational intelligence approach that will be discussed in this paper, which are the Decision Tree, SVM, and Artificial Neural Network (ANN). The data set will be normalized using the Min-Max Algorithm except for the Decision Tree approach that will be using SMOTE. As for the ANN approach, Sheela, K., Deepa, S. N., method was used to find how many hidden layers are needed for the ANN. Adam optimization technique was used iteratively in the ANN approach. A rectified linear unit (ReLu) approach was also conducted to obtain the activation function of the ANN.

I. Introduction

Water quality plays an important aspect in maintaining ecosystems around it. This includes flora, fauna and human society that lives around it. If a water body (such as river, lake, etc.) is badly polluted, the effect could danger the entire ecosystem in the water body and could danger those who live around it as they lost their livelihood (from catching fishes, etc.) and reliable water source.

To be able to determine water quality, scientists measure a variety of properties to represent the quality. By measuring the water quality firsthand, further pollution could be warned to be avoided and some actions to restore the quality could be taken. Those properties are pH value, hardness, solids, chloramines, sulfate, conductivity, organic carbon, trihaltomethanes, turbidity, and potability. Therefore, water quality of water bodies can be classified based on the ratio of the properties and from other references of other water bodies.

Around 829.000 people have died each year because of the diarrhea of consuming dirty water. Sometimes, people that live around a dirty water body did not realize that their source of water has become unpotable. This is because they have gotten used to using the water body. There are also rivers that has the characteristics that are stereotype to unpotable water body, yet in truth, the water is still potable.

The potability is hard to confirm as there is not always scientist around that are able to measure the potability. Therefore, in this paper, the property of water quality that will be determined from the other properties will be the potability. With the help of and water quality datasets (to test and train) and computational intelligence methods that will be used in this experiment, which are Artificial Neural Network (ANN) and Decision Tree, an artificial intelligence can be created to determine potability rate from the water properties and decide whether the said water is potable or not. Hopefully, anyone then could use this program to monitor potability of water body periodically so that people around the water body could be noticed before consuming unpotable water and save lives.

II. Related works

Some related works about water quality and potability prediction that has been published into paper will be discussed in this section. A Neuro-Fuzzy Inference System (WDT-ANFIS) based on augmented wavelet denoising technique was proposed. It has three techniques or assessment that were used to evaluate the models. The first one is by depending on partitioning of the neural network connection weights that marks significance. The second and the third assessment used individual parameters and a combination of parameters with a difference in scenarios that were presented for these techniques. First scenario was constructing a prediction model for water quality parameters at every station, while second scenario was developing a prediction model based on the value of the same parameter at the previous station (upstream). Both scenarios were experimented using twelve input parameters.

A model based on the principal component of regression was also proposed. The weighted arithmetic index method was used to calculate water quality index and PCA was applied to the dataset to extract the most dominant parameters. After that, regression algorithms were applied to the PCA. Afterwards, the Gradient Boosting Classifier was used to classify the water quality status. The principal component regression method achieved 95% accuracy while Gradient Boosting Classifier method achieved 100% classification accuracy.

Combination of ANN and ANN Bayesian Regularization were also used to predict water quality. By comparing the performance from the algorithm, it is indicated that the Bayesian regularization could achieve successful water quality prediction. As for the training and testing datasets, the correlation coefficients between the predicted and observed values of the water quality were 0.77 and 0.94. Sensitivity analysis has been done to demonstrate each parameter importance.

Automatic deep learning was compared with the conventional deep learning In predicting water quality. The conventional deep learning approach resulted a slightly better performance for both binary and multiclass water data. Yet, automatic deep learning found more appropriate deep learning model easier and gave better performance in the process.

A deep learning model that utilizes Long-Short Term Memory (LSTM) algorithm was proposed for IoT systems. The model was predicting water quality indicators such as salinity, temperature, pH, and dissolved oxygen for aquaculture and fisheries. The results showed that the proposed model is fit for real-world application. Additionally, monitoring the indicators of early warnings from the system could help farmers in managing water quality.

III. Dataset and Features

Water quality indicators from 3276 different water sources are included in our dataset. This dataset contains the following values: (1) pH value, when assessing the acid-base balance of water, PH is a key factor. Additionally, it shows if the water is acidic or alkaline. The maximum pH allowed range, according to WHO, it's between 6.5 and 8.5. The current investigation's ranges fell between 6.52 and 6.83, which is within WHO criteria. (2) Hardness, salts made of calcium and magnesium are the primary causes in hardness. These salts are released by the geologic formations that water passes through. How long water is exposed to a hardness-producing substance influences how hard the water is when it is in its raw state. The ability of water to form soap due to calcium and magnesium precipitation was the original definition of hardness. (3) Solids, numerous inorganic and some organic minerals or salts, including potassium, calcium, sodium, bicarbonates, chlorides, magnesium, sulfates, and others, can be dissolved by water. These minerals gave the water an undesirable flavor and faded color. This is a crucial variable while using water. Water with a high TDS rating is one that has a high mineral content. The recommended TDS level for drinking purposes is 500 mg/L, with a maximum limit of 1000 mg/L. (4) Chloramines, the two main disinfectants utilized in public water systems are chlorine and chloramine. When ammonia is added to chlorine to purify drinking water, chloramines are most frequently generated. Drinking water can include up to 4 mg/L of chlorine (or 4 ppm), which is regarded as a safe quantity. (5) Sulfate, organic compounds that are naturally present in rocks, soil, and minerals. They can be found in the surrounding air, groundwater, vegetation, and food. Sulfate is mostly used in the chemical industry for commercial purposes. The amount of sulfate in saltwater is around 2,700 mg/L. The majority of freshwater sources have values between 3 and 30 mg/L, while certain regions have substantially higher levels (1000 mg/L). (6) Conductivity, pure water is a good insulator rather than a good conductor of electrical current. The electrical conductivity of water is improved by an increase in ion concentration. The electrical conductivity of water is typically determined by the amount of dissolved particles present. The ability of a solution to convey current through its ionic process is measured by electrical conductivity (EC). According to WHO guidelines, the EC value shouldn't be more than 400 μS/cm. (7) Organic carbon, the decomposing natural organic matter (NOM) and synthetic sources both contribute to the total organic carbon (TOC) in source waters. The total amount of carbon (TOC) in organic compounds in pure water is a measurement of this. US EPA estimates that treated drinking water has < 2 mg/L of TOC and that source water, which is used for treatment, contains < 4 mg/L. (8) Trihalomethanes, can be discovered in chlorine-treated water. The amount of organic matter in the water, the quantity of chlorine needed to treat the water, and the temperature of the treated water all affect the concentration of THMs in drinking water. THM concentrations up to 80 ppm are regarded as safe for drinking water. (9) Turbidity, he amount of solid stuff present in the suspended state affects how turbid the water is. The test is used to determine the quality of waste discharge with regard to colloidal matter and measures the light-emitting capabilities of water. The Wondo Genet Campus's mean turbidity value (0.98 NTU) is less than the WHO-recommended threshold of 5.00 NTU. (10) Lastly, Potability, determines if water is safe for human consumption, with 1 denoting drinkable and 0 denoting unfit for potable.

The following is an example dataset :

Table 1 : Example datasets

	pH	*Hardness*	*Chloramines*	*Sulfate*	..
0	NaN	204.89..	7.30..	368.51..	
1	3.71..	129.42..	6.63..	NaN	
2	8.09..	224.23..	9.27..	NaN	
3	8.31..	214.37..	8.05..	356.88..	
4	9.09..	181.10..	6.54..	310.13..	

To fill the NaN, we can use the Mean Method, this means to replace all the missing data by their column's mean. We will split the dataset for this experiment into 90% training data and 10% test data (training data = 2948, test data = 328). As we observe, the data shown in table below, it is clear that the data has a big difference in mean and standard deviation.

Table 2 : Water quality properties in the datasets

Parameter	*Mean*	*SD*
ph	7.080795	1.469956
Hardness	196.369496	32.879761
Solids	22014.092526	8768.570828
Chloramines	7.122277	1.583085
Sulfate	333.775777	36.142612
Conductivity	426.205111	80.824064
Organic Carbon	14.284970	3.308162
Trihaltomethanes	66.39629	15.769881
Turbidity	3.966786	0.780382
Potability	0.390110	0.487849

That inferring that we have to normalize our data using Min-Max scaling Algorithm, to scales and translates each feature individually such that it is in the given range on the training set. With the formula :

$$(1)\ x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

IV. Methods

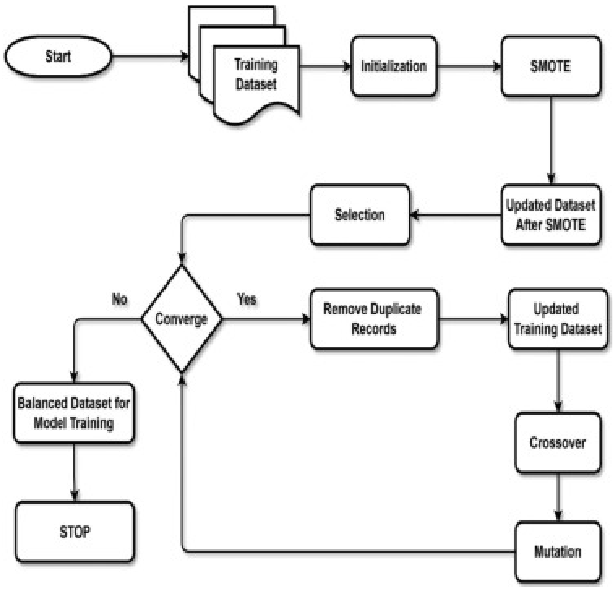
A. Decision Tree Classification

While working with the unbalanced dataset, the problem that might occur is that most machine learning models ignore the minority class, which results in poor performance, but the fact is that the minority class is often the most important class. To overcome this unbalanced dataset problem, we can use the technique of oversampling the minority class of the dataset. In this technique, replication of instances happens in the minority class which is the easiest approach, but these instances do not add much information to the model. Instead of this, we can create new instances by synthesizing old ones. 'e Synthetic Minority Oversampling Technique, or SMOTE for short, is a type of data augmentation for the minority class. SMOTE works by identifying adjacent instances in the feature space, drawing a line linking them, and generating a new sample at a position along that line. To be more precise, an instance from the minority class is chosen randomly. After analyzing the training dataset, the distribution of the training dataset is found to be non-uniform. Thus, the training dataset of the proposed system is imbalanced due to the uneven distribution of the classes for multi-class classification. Training deep learning neural network model on more data can result in more skillful and more robust models. The SMOTE augmentation techniques create variations of the data points that can improve the ability of the model to generalize the new real-time input data points. Therefore, the proposed SMOTE technique is applied only to the training dataset (70% of the original dataset) to make it a balanced dataset.

Hence, we proposed the G-SMOTE method, which hybridizes the smote technique and a genetic algorithm for handling the imbalanced dataset. The algorithm for G-SMOTE is mentioned in natural language as below.

G-SMOTE Algorithm

Step 1: Start
Step 2: Read the dataset
Step 3: Analyse the distribution of the dataset
Step 4: Initialization: initialize the water with the samples containing majority and minority samples. Also, initialize the decision variable, which sets the range of sampling
Step5: SMOTE: This step adds synthetically generated samples to the minority class using the support vector machine
Step 6: Get the updated dataset using SMOTE.
Step 7: Selection: The fitness function is developed to set the threshold for the fitness score of minority class instances. This function finds the unique and duplicate samples for each minority class.
Step 8: Check for Converge: If the probability of duplicate samples in each minority class is less than the fitness score threshold, then the dataset is said to be balanced, and the algorithm stops; otherwise, remove the duplicate samples.
Step 9: Crossover: The proposed system uses the Euclidean distance to select k- samples from each minority class instance and find the average of k-samples to generate new samples to be added.
Step 10: Mutation: It selects the mutation probability randomly in the range of 0.98–0.99 to change the sample generated by the crossover step and update the samples of minority classes.
Step 11: Go to step 8 to check for converge conditions.



G-SMOTE working mechanism.

The proposed system links the various instances (rows) of the initial training set with the sampling rate to achieve an optimum accuracy rate for classification. Mathematical modeling of the proposed G-SMOTE method is given below. The proposed system uses the genetic algorithm to achieve the best possible sampling rate for various instances, as described in equation below.

$$(2)\quad Z = G(H); \qquad \min(X) \ll Xi \ll \max\,(X)$$

$$(3)\quad H = (X1, X2, X3, \dots, Xp); \qquad i = 1, 2, 3, \dots P$$

Where Z is maximize and function G(H) is the objective function representing the accuracy rate for minority class classification and the entire dataset classification. H represents sampling rates. P represents the total count of minority class instances. Xi represents the sampling rate of minority class instance hi. minX describes the lower bound, and maxX describes the upper bound of sampling rate Xi. The proposed system uses the G-SMOTE algorithm, which hybridizes the SMOTE and the genetic algorithm to obtain the improved sampling rates to create the new balanced dataset using the oversampling method

B. *Support Vector Machine (SVM)*

Support Vector Machine is a classifying method based on the theory of statistical learning. SVM uses the structural risk minimization principle to address overfitting problems in Machine Learning by reducing the model's complexity and fitting the training data successfully. The minimization of risk can enhance the generalization of the SVM model. The estimates of the SVM model are created based on a small sub-set of training data, which is known as support vector. The capability to interpret Support Vector Machine decisions can be improved by recognizing vectors that are chosen as support vectors. SVM maps the initial data in a high-dimension feature space in which an optimal separating plane is created by using a suitable kernel function. For classification, the optimal separating plane is the line that divides the plane into two parts and each class is placed on a different side. Along each part of the separating plane, 2 parallel hyperplanes can be built to separate the training data. Let { xi, yi } * ni = 1 be the training samples, Xi ∈ Rn are the input vector and yi∈{−1,1} are the label of class. The hyperplane w * x + b = 0 where w is the weights vector, x is the input vector and b is the bias, is optimal if the margin between the closest training vector and the hyperplane is maximal. The optimal hyperplane can be constructed by solving an optimization problem as follows:

Minimize

$$\frac{1}{2}|w|^2$$

subject to

$$(4) \quad yi \cdot (w \cdot x + b) \geq 1; \;\; i = 1, 2, \dots, n$$

SVM is constructed as a dual optimization problem:

$$(5) \quad R = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \; \alpha_j \, \mathrm{y}_i \, \mathrm{y}_j \, (\mathrm{x_i} \cdot \mathrm{x}_j)$$

subject to

$$(6) \quad \sum_{i=1}^{n} \alpha_i \mathrm{y}_i = 0; \quad C \; \geq \alpha_i \geq 0$$

where R is the dual Lagrangian, C is the regularization parameter, which is employed to control the trade-off between the margin and the training error.

If αi is the Lagrange multipliers, optimal hyperplane, W can be computed as follows:

$$(7) \quad W = \sum_{i=1}^{n} \alpha_i \mathrm{y}_i \mathrm{x}_i$$

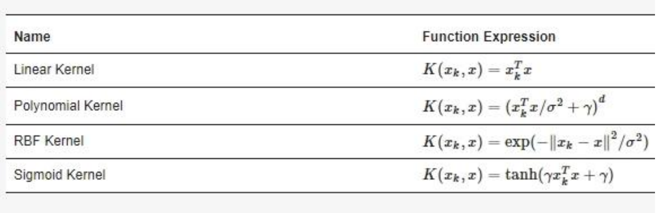
Therefore, the non-linear decision function, which is acquired by resolving the dual optimization problem, can be written as follows:

$$(8) \quad f(x) = sgn \sum_{i=1}^{n} \alpha_i \mathrm{y}_i (\mathrm{x_i} \cdot \mathrm{x}_j) + b$$

There is a kernel function k(xi·x) that allows SVM to make a non-linear classification. The value of k(xi·x) equals to φ(xi)·φ(x), where φ(·) is the transformation function that changes the input data into higher dimension feature space. Thus, the SVM non-linear decision function can be defined as follows:

$$(9) \quad f(x) = sgn \sum_{i=1}^{n} \alpha_i \mathrm{y}_i k(\mathrm{x_i} \cdot \mathrm{x}_j) + b$$

where k(xi,x) is the inner product kernel function that satisfies the Mercer conditions. There are four commonly used Mercer kernel functions, which are linear kernel, polynomial kernel, Sigmoid and Radial Basis Function kernel. Kernel function expressions are given in Table 1



Name	Function Expression
Linear Kernel	$K(x_k, x) = x_k^T x$
Polynomial Kernel	$K(x_k, x) = (x_k^T x/\sigma^2 + \gamma)^d$
RBF Kernel	$K(x_k, x) = \exp(-\lVert x_k - x \rVert^2/\sigma^2)$
Sigmoid Kernel	$K(x_k, x) = \tanh(\gamma x_k^T x + \gamma)$

***Table 1.** Kernel Functions.*

This study used a complexity constant, C=5, to set the misclassification tolerance. A high value of C can lead to overfitting problems, while a low value may cause overgeneralization. This study used the polynomial kernel since it is suitable for the case in which all training data are normalized

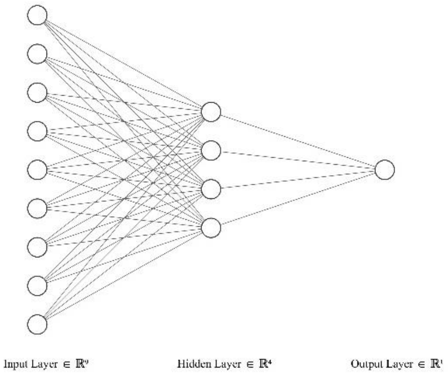
C. Artificial Neural Network (ANN)

In this experiment, we will use Artificial Neural Network. Artificial neurons, which are a set of interconnected units or nodes that loosely mimic the neurons in a biological brain, are the foundation of ANN. As shown below, 9 neurons reflect the nine explanatory variables that are crucial for predicting water quality in the input layer of the ANN. To find how many the hidden layer needed for the Artificial Neural Network, we used Sheela, K., Deepa, S. N., method, (Yotov, 2020) Sheela and Deepa’s method has been the most successful in predicting the optimal number of neurons. The method has a formula:

$$(10) \quad N_h \; = \; \frac{4N_i^2 \; + \; 3}{N_i^2 \; - \; 8}$$

Where N_h denotes number of optimal hidden neurons, and N_i denotes number of input neurons. This method find the optimal correct hidden neurons by selecting criteria for fixing hidden neuron. Then training the network and evaluate the performance of network, therefore it can calculate it’s perfomance result such as output and prediction error. The least prediction error will denotes the correct criteria (formula) to deduct the optimal hidden neuron. The results show that the proposed work produced superior outcomes to the other alternatives. In this study, recommended standards for developing three-layer neural networks are taken into account. It is well known that some methods result in larger networks than necessary while others are expensive. From the observation, the formula is really effective for prediction in renewable energy systems. Finally, there is only one neuron in our output layer because there is only one probability of water quality in the output. So the Artificial Neural Network (ANN) architecture will be;

Picture 1: Artificial Neural Network

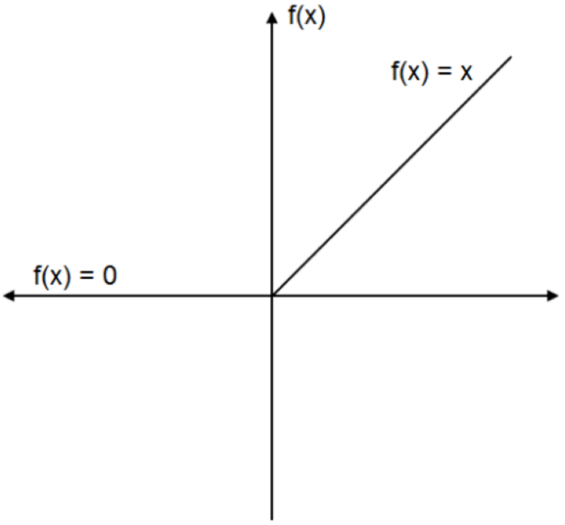


For the optimizer algorithm, we used Adam. In place of the conventional (SGD) stochastic gradient descent method, Adam is an optimization technique that may be used to iteratively update network weights depending on training data. The reason for using this method, is because Adam is well suited for problems that are large in terms of data and/or parameters. The comparison from using SGD, for all weight updates, stochastic gradient descent maintains a constant learning rate (referred to as alpha), which does not change throughout training. As learning progresses, a learning rate is maintained and independently adjusted for each network weight (parameter). According to the authors, Adam combines the benefits of two additional SGD's modifications. Specifically : (1) Adaptive Gradient Algorithm (AdaGrad) keeps a per-parameter learning rate that enhances performance, (2) Root Mean Square Propagation (RMSProp) maintains adaptive per-parameter learning rates depending on the weight gradients' average recent recent magnitudes (e.g. how quickly it is changing). The algorithm performs effectively on online and non-stationary issues, according to this (e.g. noisy). Adam uses the average of the second moments of the gradients in addition to the average of the first moments, which is how RMSProp adjusts the parameter learning rates (the uncentered variance). The parameters *beta1* and *beta2* control the decay rates of these moving averages, and the algorithm specifically creates an exponential moving average of the gradient and the squared gradient. Moment estimations are biased towards zero as a result of the initial value of the moving averages and *beta1* and *beta2* values that are near to 1.0 (recommended). By first calculating the biased estimates and then the bias-corrected estimates, this bias is eliminated.

For the activation function, we used ReLu, a rectified linear unit (also known as a unit using a rectifier) produces raw output if the input is greater than 0, and 0 otherwise. In other words, the output is identical to the input if the input is higher than 0. ReLU's operation is more similar to how organic neurons function.

$$(11)\quad f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$$

Picture 2: Graph of organic neurons function



Using ReLU allows one to activate fewer neutrons, which improves the performance of the entire network.

Lastly, the selection of loss function is an essential next step in ANN modeling. The loss function computes the difference between each neuron's actual and target values, calculating the prediction accuracy. Up until the global minimum error is attained, the network is trained. The loss function was calculated using the root mean square error (MSE) :

$$(12)\quad MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i' - y_i\ \right)^2$$

V. Experiments/Results/Discussion

A. Decision Tree Classification Approach

We try some approach to testing this data to ensure our analysis. One of approach that we use is Decision tree classification approach because procedure of decision tree classification and ANN is different. First, we normalized the data by SMOTE function. Afterward, we conduct analysis best parameters. We used optuna for this analysis. We get the best trial for params are 'max_depth' is 7, 'min_samples_leaf' is 1, 'min_samples_split' is 24. We also get best values, that is 0.63 at first time we run the code. Then, we conduct Cross validation and we get result as given in the table below.

Table 2: Cross-validation ANN

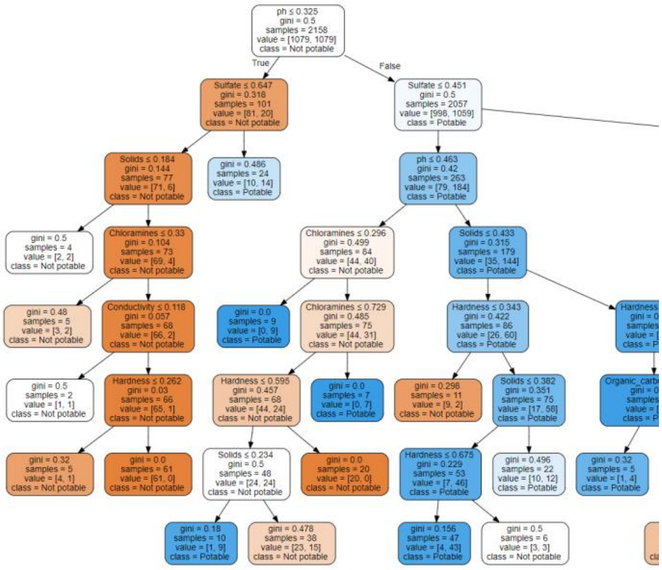
	Precission	recall	F1-score	support
0	0.82	0.59	0.69	126
1	0.54	0.79	0.64	76
accuracy			0.66	202
Macro avg	0.68	0.69	0.66	202
Weighted avg	0.71	0.66	0.67	202

ROC AUC score is 0.783. Cross-validation scores with 10 folds is given in table below

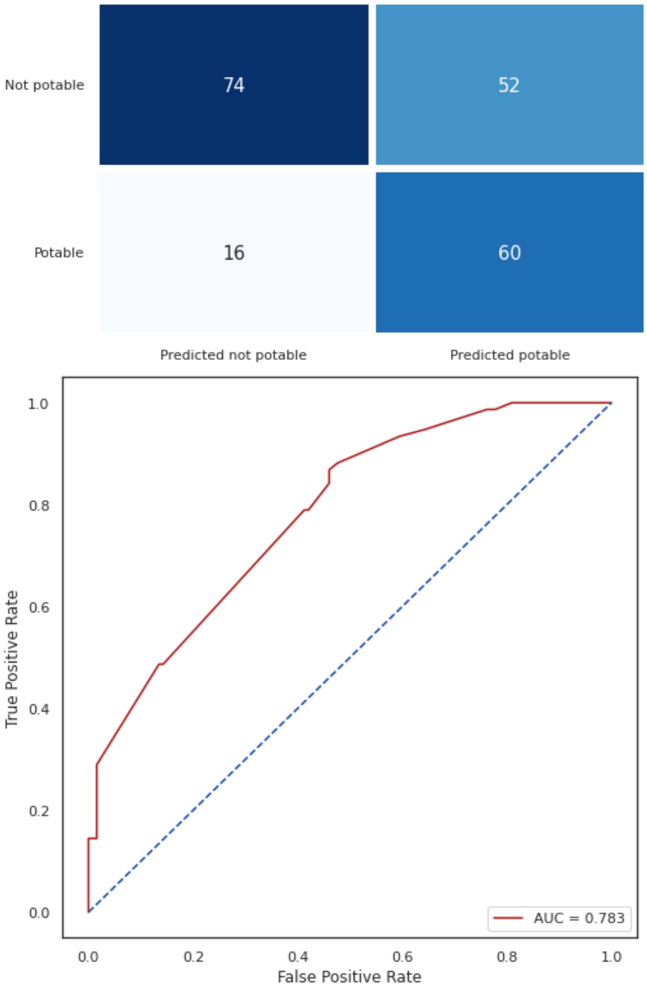
Table 3: Cross-validation scores with 10 folds

Parameter	*Value*
ROC AUC	0.676
Precision	0.63
Recall	0.63
F1	0.62

After we get the result, we conduct visualize the decision tree. The node of the decision tree is a lot. Image below is image snippet of decision tree visualization.


ph ≤ 0.325 gini = 0.5 samples = 2158 value = [1079, 1079] class = Not potable
True
False
Sulfate ≤ 0.647 gini = 0.318 samples = 101 value = [81, 20] class = Not potable
Sulfate ≤ 0.451 gini = 0.5 samples = 2057 value = [998, 1059] class = Potable
Solids ≤ 0.184 gini = 0.144 samples = 77 value = [71, 6] class = Not potable
gini = 0.486 samples = 24 value = [10, 14] class = Potable
ph ≤ 0.463 gini = 0.42 samples = 263 value = [79, 184] class = Potable
gini = 0.5 samples = 4 value = [2, 2] class = Not potable
Chloramines ≤ 0.33 gini = 0.104 samples = 73 value = [69, 4] class = Not potable
Chloramines ≤ 0.296 gini = 0.499 samples = 84 value = [44, 40] class = Not potable
Solids ≤ 0.433 gini = 0.315 samples = 179 value = [35, 144] class = Potable
gini = 0.48 samples = 5 value = [3, 2] class = Not potable
Conductivity ≤ 0.118 gini = 0.057 samples = 68 value = [66, 2] class = Not potable
gini = 0.0 samples = 9 value = [0, 9] class = Potable
Chloramines ≤ 0.729 gini = 0.485 samples = 75 value = [44, 31] class = Not potable
Hardness ≤ 0.343 gini = 0.422 samples = 86 value = [26, 60] class = Potable
Hardness gini = 0 samples value = [class = P
gini = 0.5 samples = 2 value = [1, 1] class = Not potable
Hardness ≤ 0.262 gini = 0.03 samples = 66 value = [65, 1] class = Not potable
Hardness ≤ 0.595 gini = 0.457 samples = 68 value = [44, 24] class = Not potable
gini = 0.0 samples = 7 value = [0, 7] class = Potable
gini = 0.298 samples = 11 value = [9, 2] class = Not potable
Solids ≤ 0.382 gini = 0.351 samples = 75 value = [17, 58] class = Potable
Organic_carb gini = 0 samples value = [class = P
gini = 0.32 samples = 5 value = [4, 1] class = Not potable
gini = 0.0 samples = 61 value = [61, 0] class = Not potable
Solids ≤ 0.234 gini = 0.5 samples = 48 value = [24, 24] class = Not potable
gini = 0.0 samples = 20 value = [20, 0] class = Not potable
Hardness ≤ 0.675 gini = 0.229 samples = 53 value = [7, 46] class = Potable
gini = 0.496 samples = 22 value = [10, 12] class = Potable
gini = 0.32 samples = 5 value = [1, 4] class = Potable
gini = 0.18 samples = 10 value = [1, 9] class = Potable
gini = 0.478 samples = 38 value = [23, 15] class = Not potable
gini = 0.156 samples = 47 value = [4, 43] class = Potable
gini = 0.5 samples = 6 value = [3, 3] class = Not potable
cla

Picture 5: Confused matrix and ROC Curve



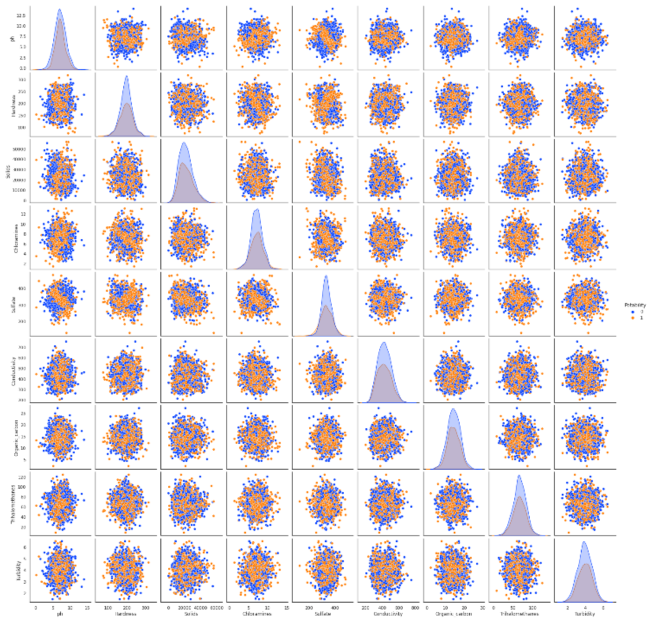
(a) Confused Matrix

(b) ROC Curve

B. *Support Vector Machine (SVM) Approach*

Firstly, we visualize the data. Image below is a visualization of the initial data.

Picture 6: Data visualization



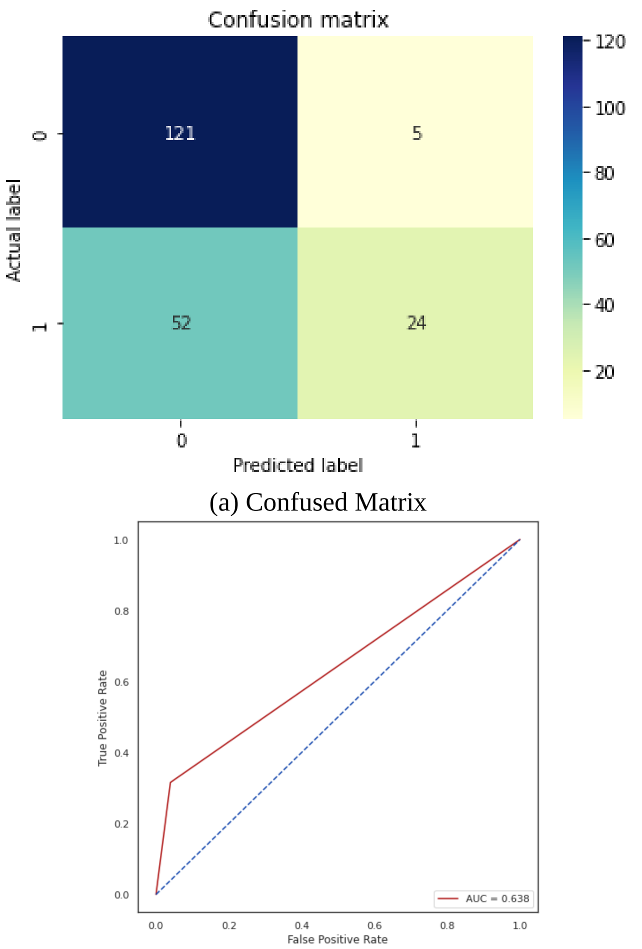
Secondly, we conduct analysis the data using SVM approach. The SVM approach was conducted using python svm library from sklearn with max_iter set equal to 1, which means that the iteration will have no limit and will be terminated automatically by the machine when the machine found the appropriate time for termination. After that, we predicted the test and we get the result as in table below

Table 4: Result of SVM approach

Model	Acc	Precision	Recall	F1
SVM	0.7178	0.7476	0.7178	0.6768

The following picture shows confused matrix and ROC Curve of this approach.

Picture 7: Confused Matrix and ROC Curve of SVM Approach



(a) Confused Matrix

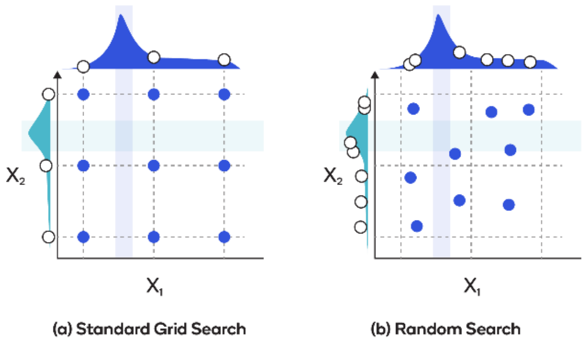
(b) ROC Curve

C. *Artificial Neural Network (ANN) Approach*

For our selection of hidden neurons layers = 4, the accuracy is relatively low 0.62. So we have to do trial and error to find the optimum number of hidden neurons, besides that we can tune in more on the types of optimizer algorithm and activation function. To find the optimal method and number of neurons, in this experiment we use Grid Search algortihm to find the best option with highest accuracy.

Grid Search involves setting up a grid of hyperparameters and training/testing our model on all of the potential combinations. We can now look at the parameters that worked best with Random Search to determine the ones to utilize in Grid Search and create a grid based on those parameters to see if a better combination can be found.

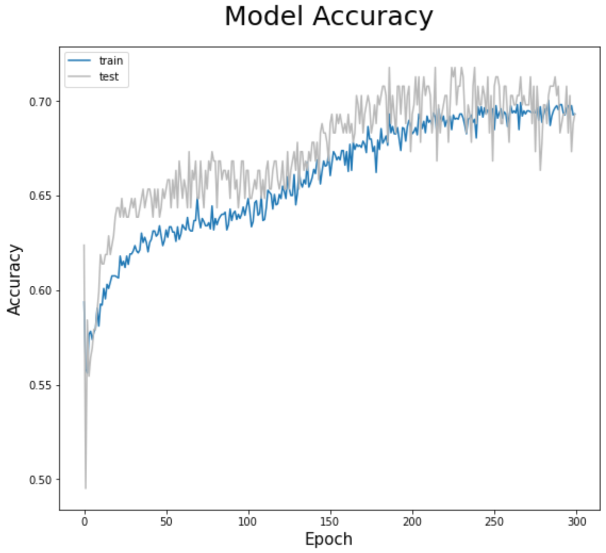
Picture 3: Standard Grid Search and Random Search



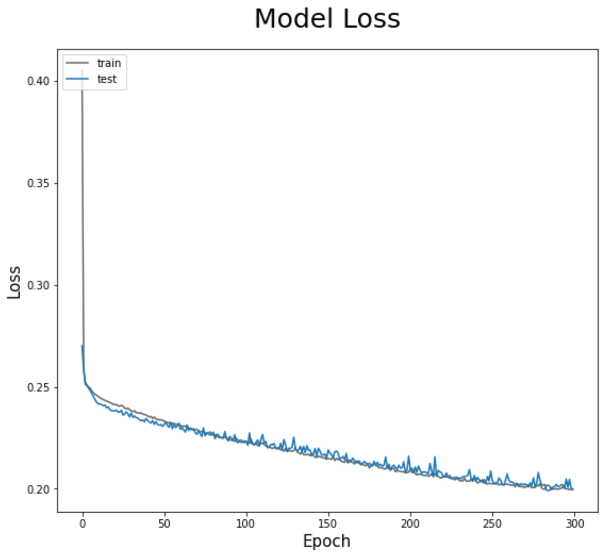
src : [*https://www.kaggle.com/code/pavansanagapati/automated-hyperparameter-tuning*](https://www.kaggle.com/code/pavansanagapati/automated-hyperparameter-tuning)

After running Grid Search on the parameters, we found out that neuron 20, activation ReLu, and optimizer Adam. With these parameters, our Artificial Neural Network model produces better results with average accuracy 0.68397. After trial and error, the optimum neurons for the problem is 20, which several paper also used to conduct water quality including Vijay, 20 neurons were the ones that outperformed the other methods in various experiments on water quality.

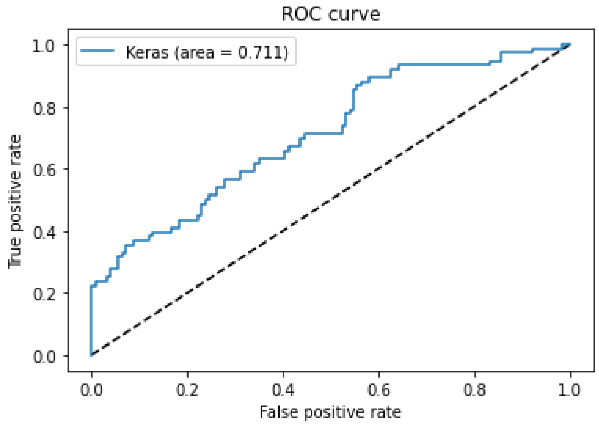
Picture 4: Model Accuracy, Model Loss, ROC Curve, and Confusion Matrix Graph



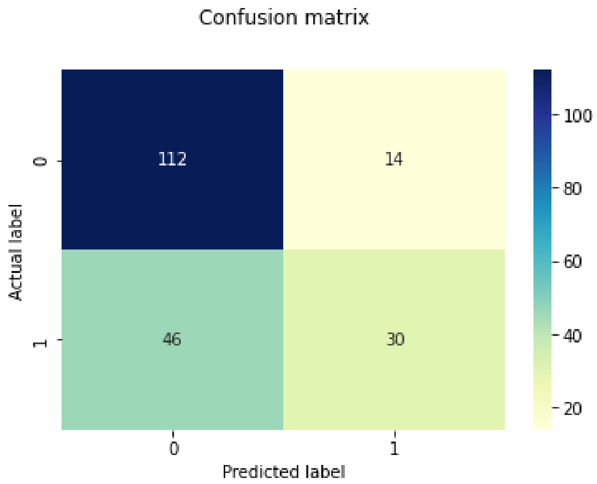
(a) Model Accuracy



(b) Model Loss



(c) ROC Curve



(d) Confusion Matrix

In this section, the main primary metrics is precision. We then conduct cross-validation, which result in

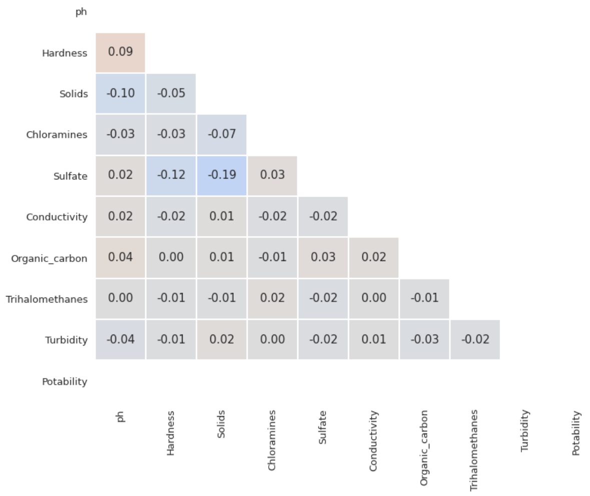
Table 3: Cross-validation Result

Parameter	*Value*

ROC AUC	0.713
Precision	0.682
Recall	0.622
F1	0,658

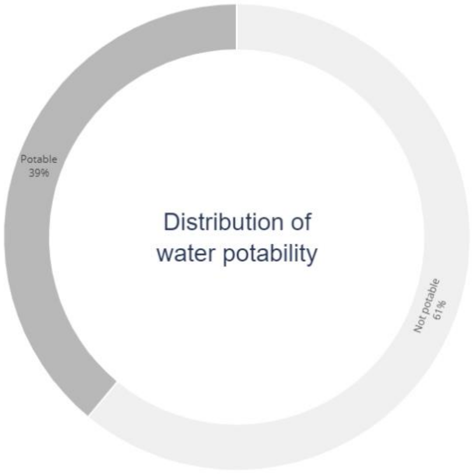
Lastly, we will analyze how can we get those results from the dataset. Let's start by looking at how the current qualities correlate with one another. There is little association, as can be seen in the heatmap below.

Picture 4: Heatmap



From the datasets, we can deduct how the ‘Water Potability’ distribution along the datasets, so this dataset tends to have data with Water that is Not Potable

Picture 5: Distribution of water potability



These findings show a really interesting conclusion, which raises several assumption from us regarding the not-so-high accuration. The amount of many indicators, including solids, chloramines, sulfates, and organic carbon, is hazardous in all observations. It may not possible for these indicators to have higher values for drinkable water. The indicators should be attributed and/or biased to non-potable water based on basic principles of chemistry. The only plausible reason we can find a drinkable water result is that the normal signs in one category would level out as a result of the heightened indicators in another area, but this is may not be the case. From the heatmap above, it shows also that the distribution of potable and non-potable water in every column is nearly identical.

VI. Conclusion/Future Work

The algorithm that we have used in this experiment are Decision Tree, SVM and ANN. The model produced from the Decision Tree gave 63% accuracy and AUC of 0.783. As for the result of SVM experiment, it is found that the model have 71.78% accuracy and 0.638 AUC. This means that SVM model gives better prediction than the model from Decision Tree approach.

The ANN model for hidden neurons layers of 4 gave accuracy that is relatively low, which is 0.62. For that reason, we have performed some trials and errors to seek out the optimum range of hidden neurons. When performing grid search algorithm, we also used somatic cell twenty, activation ReLu, and optimizer Adam. With those methods, our ANN model produced higher results with average accuracy of 68.4% and AOC of 0.549. From the trial and error, it is found that the optimum neurons are 20. Yet, 20 neurons were those that outperformed the opposite ways in numerous experiments on water quality.

From those 3 experiments, it is obviously found that SVM approach gave the best result with the accuracy of 71.78%. This accuracy still tolerable for an artificial intelligence model, which is above 70%. Yet, it is still considered rather low to be used for real world applications. As for the other two methods, both of them gave rather low accuracy and is below 70%, which means that the models obtained are considered unable to predict the water potability.

These low results findings from Decision Tree and ANN approach triggered some subjective assumptions from us concerning the not-so-excessive accuration. The quantity of many signs, inclusive of solids, chloramines, sulfates, and natural carbon, is unsafe in all observations. It will not be viable for those signs to have better values for drinkable water. However, that is will not the objective answer to the inquiry. From the heatmap above, it indicates additionally that the distribution of potable and non-potable water in each column is sort of low, which means the relations between the properties regarding potability is low.

As for why the SVM model gave better performance than Decision Tree and ANN is because SVM could process linearly or non-linearly separable data better. ANN minimizes only the empirical risk learnt from the training samples, but SVM considers both this risk and the structural risk. SVM could also send the data to a higher dimensional space where it is linearly separable. The data set used in this experiment is also rather small and SVM provides better accuracy to smaller data sets than ANN. Also, the kernel trick that is used in SVM has the computational advantage that is usually much faster to compute a single non-linear function than to pass the vector through many hidden layers in ANN. Thus, the SVM model gave the better result.

For our next project, we will try to find another data set that has higher heatmap values and larger sets. When we have found the correct data set for this experiment, we could decide what are the properties that are important in indicating potability of a water.

VII. Contributions

Zahra Fayyadiyati has conceived the presented idea, write the abstraction, find and summarize related works, code the project, conduct the experiment and write the conclusion in Chapter VI. Wina Tungmiharja has found the dataset, understanding the algorithm to process the dataset, code the project, conduct the experiment and also write the theory in this article, that is Artificial Neural Network (ANN). Nazhifah Elqolby has presented theory about SVM, Decision Tree, and G-SMOTE algorithm. Nazhifah also conceived the conclusion. Luthfiyyah H. Amari has presented about experimental of SVM and Decision tree.

Google Colab Source Code:
https://colab.research.google.com/drive/1RSNd8iqaVGGIlWfH_0WceaDVtv8cpTMx?usp=sharing

References

Eason G., Noble B. and Sneddon Ian Naismith. (1955). On certain integrals of Lipschitz-Hankel type involving products of bessel functions. *Philosophical Transactions of the Royal Society of London*. *Series A, Mathematical and Physical Sciences.* 247: 529–551. http://doi.org/10.1098/rsta.1955.0005

Kadiwal, A. (2021). Water Quality : Drinking water potability, Version 1. Retrieved November 14, 2022 from https://www.kaggle.com/datasets/adityakadiwal/water-potability

Estrada, I. (2022). Water Quality Prediction Using ANN, Version 1. Retrieved November 14, 2022 from https://www.kaggle.com/code/irvingestrada/water-quality-prediction-using-ann/notebook

Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research,* vol.12, pp. 2825-2830. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Yotov, K., Hadzhikolev, E., & Hadzhikoleva, S.(2020). Determining the number of neurons in artificial neural networks for approximation, trained with algorithms using the Jacobi matrix. *TEM Journal*, 9(4), 1320.

Sheela, K. G., & Deepa, S. N. (2013). A new algorithm to find number of hidden neurons in Radial Basis Function Networks for wind speed prediction in renewable energy systems. Journal of Control Engineering and Applied Informatics, 15(3), 30-37.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Vijay, S., & Kamaraj, K. (2021). Prediction of water quality index in drinking water distribution system using activation functions based Ann. Water Resources Management, 35(2), 535-553.

R. Zhao. (2021). The Water Potability Prediction Based on Active Support Vector Machine and Artificial Neural Network, *2021 International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR)*, 2021, pp. 110-114, doi: 10.1109/ICBAR55169.2021.00032.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

G. Douzas, F. Bacao, and F. Last, “Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE,” Information Sciences, vol. 465, pp. 1–20, 2018