Machine Learning Model for Water Quality Classification

Oni Luca
Computer Science Department
American College of Thessaloniki
Thessaloniki, Greece
20200008@student.act.edu

***Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. **CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)**

Keywords—artificial intelligence, machine learning, water quality, deep learning neural networks

I. INTRODUCTION

Water is a necessary resource for all living things. Quality drinking water is critical, as contaminated or poor-quality water can cause serious health problems. Presences of minerals such as calcium, magnesium, sodium, and potassium make up essential parts of water quality. Certain minerals are necessary for human health, but too much or too little might cause problems. As a result, it is critical to monitor and categorize water depending on its mineral composition.

Conventionally, we use chemical analysis techniques to assess water quality. Nevertheless, these procedures are time-consuming, necessitate resources, and may not be suitable for on-site testing. Recent breakthroughs in machine learning and deep learning neural networks have expanded the scope of water quality assessments. Deep learning neural networks, in particular, have shown promise in their ability to categorize water samples based on mineral composition.

In this paper, we will take a gander into the use of deep-learning neural networks for classifying drinkable water based on the minerals it contains. In comparison to traditional chemical analysis techniques, we will investigate the ability of deep learning models to accurately classify water samples based on their mineral content. We will also look at the benefits and drawbacks of using deep learning neural networks for water quality analysis, as well as discuss the implications of our findings for water quality monitoring and management.

Overall, the goal of this study is to add to the growing body of knowledge about the use of deep-learning neural networks for water quality analysis. We can ensure the safety and well-being of communities that rely on clean drinking water by developing accurate and efficient methods for water quality analysis.

II. DATASET AND EXPLANATION

The dataset "Water Quality" **[4]** available on Kaggle is a collection of simulated data on water quality in an urban environment. This dataset includes attributes related to mineral content in water, such as aluminum, ammonia, arsenic, barium, cadmium, chloramine, chromium, copper, fluoride, bacteria, viruses, lead, nitrates, nitrites, mercury, perchlorate, radium, selenium, silver, uranium, and a class attribute called is safe, which is either 0 or 1 indicating whether the water sample is safe for drinking. This dataset contains only numeric variables, and each attribute has a dangerous threshold above which the water is considered unsafe for human consumption. This dataset is created for educational purposes, practice, and gaining the necessary knowledge in the field of water quality analysis.

-Table depicting each column in dataset that contribute to the classification of water

Attribute	Condition for water to be dangerous to drink
Aluminum	>2.8
Ammonia	>32.5
Arsenic	>0.01
Barium	>2
Cadmium	>0.005
Chloramine	>4
Chromium	>0.1
Copper	>1.3
Fluoride	>1.5
Bacteria	>0
Viruses	>0
Lead	>0.015
Nitrates	>10
Nitrites	>1
Mercury	>0.002
Perchlorate	>56
Radium	>5
Selenium	>0.5
Silver	>0.1
Uranium	>0.3

-Difficulties regarding the dataset

While the dataset on water quality provided at the given link is valuable for educational and practice purposes, we need to consider some limitations. **[3]**

- The dataset is simulated rather than based on real-world water quality data. As a result, the data's accuracy may not reflect real-world scenarios and may be biased toward a specific context or location. It is also highly imbalanced, with a 4:1 ratio of drinkable to undrinkable water.
- The dataset only contains information on 21 water quality attributes, and other important factors such as pH levels, total dissolved solids, or organic contaminants may impact water safety.
- There is no information in the dataset about the source of the water samples or the treatment processes they go through before being tested. That can make it difficult to make accurate predictions and draw conclusions about water quality in various settings and contexts.
- Finally, the dataset contains no missing or null values, a common problem in real-world datasets.

If we don't consider these factors, this can lead to overfitting or incorrect analysis.

-Metrics

Performance metrics such as per class recall (accuracy), per class precision, average accuracy, and average F1 score are commonly used in classification tasks, including those related to water quality analysis.

The proportion of correct predictions made for each class is measured by the metric: Per Class Recall or Accuracy. For each class, we calculate the ratio of true positive predictions to the sum of true positive and false negative predictions. This metric indicates how well a model can identify a particular class of interest in a dataset.

$$\boldsymbol{Recall\ =\ true\,positives\ /\ (true\,positives\ +\ false\,negatives)}$$

The proportion of true positive predictions made for each class, on the other hand, is measured by Per Class Precision. For each class, it is calculated as the ratio of true positive predictions to the sum of true positive and false positive predictions. This metric provides insight into how well the model performs when identifying a class.

$$\boldsymbol{Precision\ =\ true\,positives\ /\ (true\,positives\ +\ false\,positives)}$$

Average Accuracy is the average of per-class accuracy values and provides an overall measure of how well the model classifies data.

$$\boldsymbol{Accuracy\ =\ (sum\,of\,true\,positives\,across\,all\,classes)\ /\ (sum\,of\,true\,positives\ +\ false\,negatives\,across\,all\,classes)}$$

The weighted average of the precision and recall values for each class is used to calculate the average F1 Score. It has a value between 0 and 1, with higher numbers indicating better performance. It is an accurate measure of overall model performance that takes into account both false positives and false negatives. When there is a class imbalance in the data, the F1 score can help balance the trade-off between precision and recall.

*F1 Score = 2 * (precision * recall) / (precision + recall)*
Average F1 Score = (sum of F1 scores across all classes) / (number of classes)

Overall, these metrics provide a quantitative way to evaluate a model's performance in classifying water quality data and can help inform water quality management and monitoring decision-making.

III. PROPOSED MODEL

-Model

Multiple layers of interconnected neurons that process input data and predict output classes make up a DNN model.

The concentrations of various minerals and other attributes of the water samples are likely to be input features in the context of water quality classification. DNN model procceses the input through a series of hidden layers where each layer contains multiple neurons with varying activation functions and weights. The model's output layer is typically made up of neurons corresponding to the number of output classes, in this case, safe or not safe.

Backpropagation, a technique that computes the gradient of the loss function with respect to the model parameters, is used by the DNN model during training to adjust its weights and biases. Through minimizing the difference between the predicted outputs and the actual labels of the training data, the model gradually improves its predictions over time. **[5]**

Various hyperparameters, such as the number of layers, the number of neurons in each layer, the learning rate, and the dropout rate, can be tuned to improve the performance of the DNN model. Regularization techniques like L2 regularization and weight decay are also be used to prevent overfitting and improve the model's ability to generalize to new data.

Overall, DNN models are powerful tools for the classification of complex datasets such as water quality, and they can achieve high accuracy and generalization performance with proper tuning and optimization.

-Model parameters

Several hyperparameters are critical for the performance of the deep neural network (DNN) model that we train for the classification task of drinkable water based on the minerals contained in it. Dropout Rate, Input Features, Layer Neurons, Classes, Max Epoch, Batch Size, Learning Rate, RegularizeL2, and Weight Decay are the parameters.

1) Dropout Rate: Dropout is a regularization technique used in deep neural networks to prevent overfitting. While training, the dropout rate randomly removes some neurons, forcing the model to learn more robust features. The dropout rate is the likelihood that the model removes a given neuron during training.

2) Input Features: The attributes or variables used as input to the DNN model are referred to as input features. The various mineral concentrations measured in water samples are likely to be the input features in the case of water quality classification.

3) Layer Neurons: The layer neuron is the number of neurons in each layer of the DNN model. The optimal number of neurons in a layer is determined by the complexity of the problem being solved, the amount of data available, and other model hyperparameters.

4) Classes: The number of output classes in the classification problem is denoted by the term "classes." There are two types of drinkable water classification: safe and unsafe.

5) Max Epoch: This is the maximum number of epochs or iterations of the training process. During training, an epoch is one complete iteration of the entire dataset through the DNN model.

6) Batch size: The number of samples processed by the model in one training iteration is referred to as the batch size. Smaller batch sizes may result in faster training times, but larger batch sizes may result in better generalization performance.

7) Learning Rate: Determines the step size at each iteration of the stochastic gradient descent optimization algorithm used to train the DNN model. A faster learning rate can lead to faster convergence, but it can also cause instability and poor generalization.

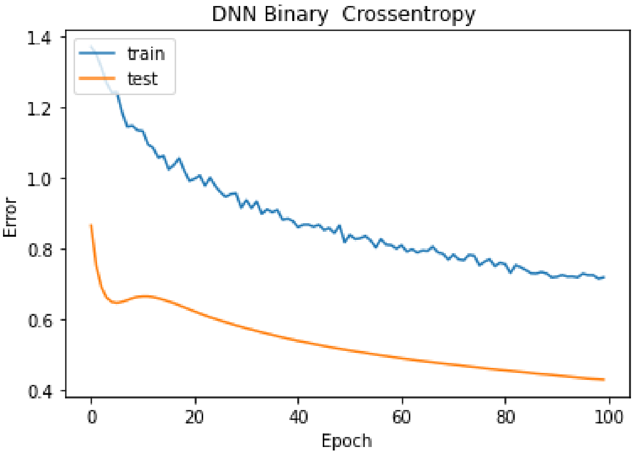
8) RegularizeL2 and Weight Decay: Regularization techniques such as L2 regularization and Weight Decay are used to prevent overfitting in DNN models during training by adding a penalty term to the loss function. These settings dictate how much regularization is applied to the model during training.

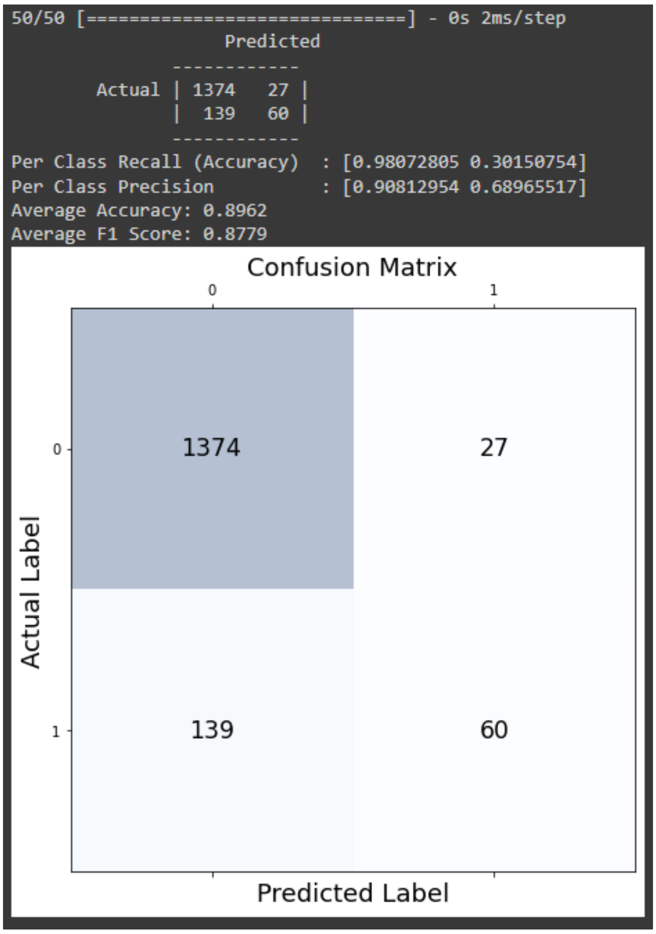
9) Weight Class: The weight_class metric can be used to tackle the problem of imbalanced datasets. It assigns a weight to each class based on its occurrence in the dataset. This weight is then used to modify the loss function during model training, thereby placing greater emphasis on the minority class.

In summary, each of these parameters is critical in the training and optimization of a deep neural network model for the mineral content classification of drinkable water. Proper tuning of these parameters can greatly improve the model's performance and ability to make accurate predictions.

-Results from baseline model

The results from the baseline, as presented below, had an accuracy rate of 89% and F1-score of 88%. However, it lacks in per-class accuracy and precision. The confusion matrix depicts a problem as the model is predicting non-drinkable water as drinkable and the model suffers as the dataset is heavily unbalanced.





```
50/50 [==============================] - 0s 2ms/step
                    Predicted
                -----------
        Actual | 1374   27 |
               |  139   60 |
                -----------
Per Class Recall (Accuracy)  : [0.98072805 0.30150754]
Per Class Precision          : [0.90812954 0.68965517]
Average Accuracy: 0.8962
Average F1 Score: 0.8779
```

As we can see the confusion matrix is not up to par, which brings us to the improved model.

IV. Improved Model

After continuous trial and error, the changes to the model configuration is as follows:

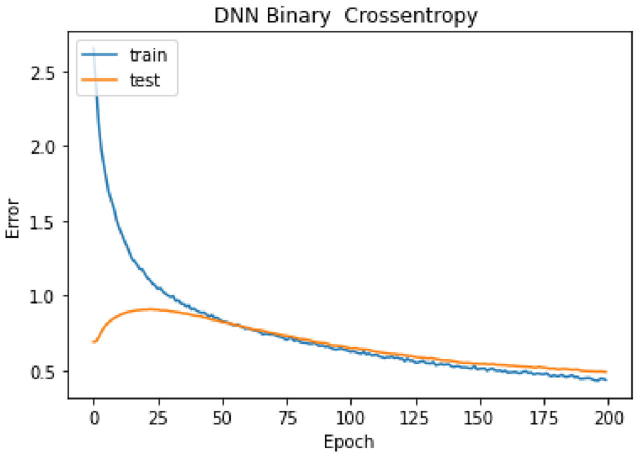
Model_config	Baseline_Model	Improved_Model
DNN-Layer	[24, 32, 32, 1]	[32,32,16,16,16, 1]
Epochs	100	200
BatchSize	200	128
L_Rate	0.0001	0.0005
Weight_decay	None	1e-4

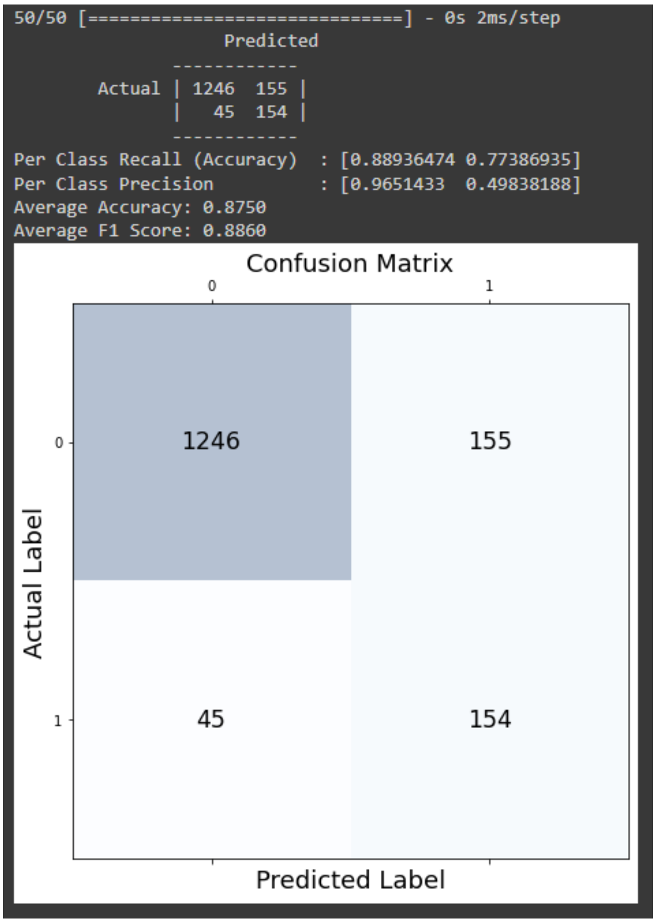
While the new epoch number is doubled, the batch size is lowered to increase performance. The learning rate is also increased and the hidden layers are added to create a deeper neural network. We also introduced regularization techniques Regulize_L2 and weight decay which were used to prevent over-fitting.

Another improvement which had a big impact in training is the usage of SMOTE technique in order to increase the under-sampled class attribute of 1. **[2]**

-Results from improved model (SMOTE TECHNIQUE)

While there is a slight downgrade in accuracy by 1.5%, we see a rise in per-class accuracy which is what we are striving for.

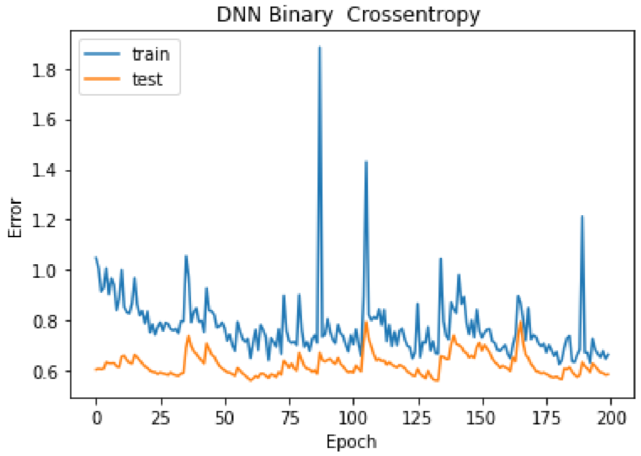


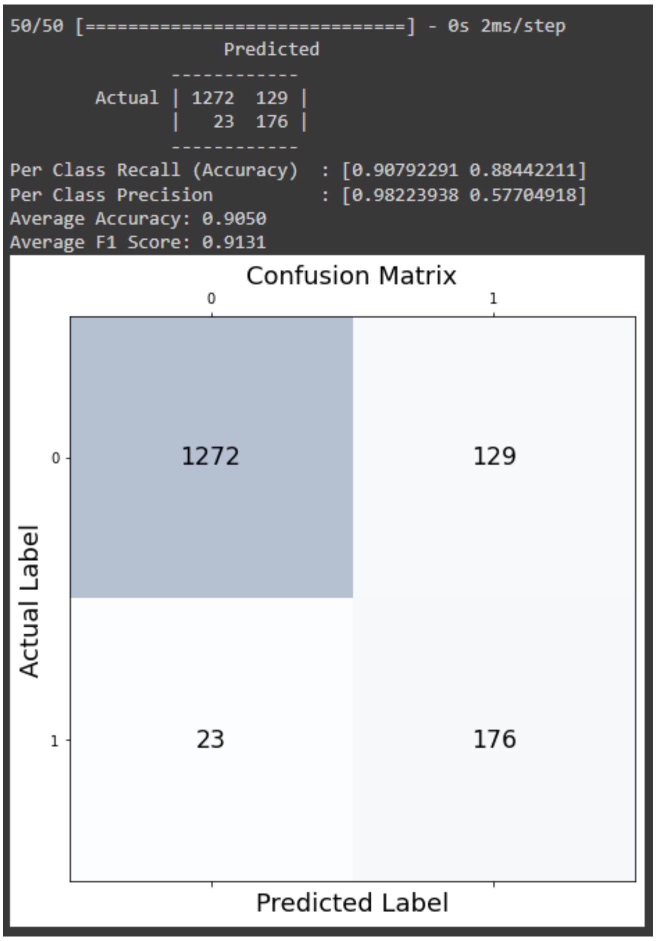


```
50/50 [==============================] - 0s 2ms/step
                        Predicted
                 ------------
        Actual | 1246  155 |
               |   45  154 |
                 ------------
Per Class Recall (Accuracy)  : [0.88936474 0.77386935]
Per Class Precision          : [0.9651433  0.49838188]
Average Accuracy: 0.8750
Average F1 Score: 0.8860
```

-Results from improved model (Weight_Class Metric)

After some deliberation, the SMOTE technique is dropped and more importance was given to the weight_class metric. Multiple ratio were tried and the best ratio proved to be [0:1, 1:1000] as the class 1 is heavily imbalanced. Here are the results from changing the weigth_class metric in the fit function provided by KERAS.





```
50/50 [==============================] - 0s 2ms/step
                        Predicted
                 ------------
        Actual | 1272  129 |
               |   23  176 |
                 ------------
Per Class Recall (Accuracy)  : [0.90792291 0.88442211]
Per Class Precision          : [0.98223938 0.57704918]
Average Accuracy: 0.9050
Average F1 Score: 0.9131
```

As we can see we have a greater accuracy rate and F1 score, however the metric we were looking to improve, per_class_recall, has an increase of 11% which is great since there is a higher chance of better classifying drinkable water.

V. Future improvements

Here are some future improvements I want to look more into.

1. Data augmentation: Adding synthetic data to the dataset can help the model generalize better. Random transformations to existing data, such as rotations, translations, and distortions, can help us accomplish data augmentation.

2. Model architecture: Experimenting with different architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), may improve model accuracy. Furthermore, incorporating pre-trained models such as transfer learning can help to leverage existing knowledge and accelerate training.

3. Hyperparameters tuning: Fine-tuning hyperparameters such as the learning rate, the number of layers, and the dropout rate can help to improve the model's performance. Techniques such as grid search, random search, or Bayesian optimization can also improve the performance.

4. Handling imbalanced data: If the dataset is imbalanced, with one class having significantly fewer samples than the other, we can use oversampling or under sampling techniques to balance the dataset and improve the model's accuracy.

5. Ensembling: Using an ensemble of multiple DNN models can help improving the accuracy of the model and reduce overfitting. We can use techniques such as bagging or boosting to achieve ensembling.

VI. Conclusions

We concluded that DNN models are an effective tool for accurately and efficiently classifying water based on mineral content. The model's performance was further improved by fine-tuning hyperparameters such as the learning rate, number of layers, and dropout rate. L2 regularization and weight decay are two regularization techniques used to prevent overfitting and improve the model's generalization performance. Additional enhancements, such as data augmentation, model architecture selection, and ensembling, can also improve model accuracy.

When discussing the importance of this research, this study can provide significant value in multiple ways. Water treatment facilities can use a more efficient and accurate method of monitoring the water quality, thus stopping water-borne diseases and illnesses.

Finally, this research can also help developing advanced machine-learning models that can be applied to other areas of water quality testing and monitoring, leading to further advancements in water treatment and public health.

References

[1] Amidi, A. and Amidi, S. (2022) *CS 230 - Deep Learning Tips and Tricks Cheatsheet* [Online]. Available at https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-deep-learning-tips-and-tricks.

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002) ‘SMOTE: Synthetic Minority Over-sampling Technique’, *Journal of Artificial Intelligence Research*, vol. 16, no. 16, pp. 321–357 [Online]. DOI: https://doi.org/10.1613/jair.953.

[3] Huang, X., Kroening, D., Ruan, W., Sharp, J., Sun, Y., Thamo, E., Wu, M. and Yi, X. (2020) ‘A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability’, *Computer Science Review*, vol. 37, p. 100270 [Online]. DOI: https://doi.org/10.1016/j.cosrev.2020.100270 (Accessed 11 December 2020).

[4] MSSMARTYPANTS (2021) *Water Quality Dataset* [Online]. Available at https://www.kaggle.com/datasets/mssmartypants/water-quality. (Accessed 18 March 2023).

[5] Zhang, A., Lipton, Z. C., Li, M. and Smola, A. J. (2021) *Dive into Deep Learning — Dive into Deep Learning 0.16.0 documentation* [Online]. Available at https://d2l.ai/.