Multi-Class Classification of Obesity Level using an Artificial Neural Network

John Michael S. Tugay

***Abstract* – Current health crises experienced by the world led to less access of people to healthcare monitoring, and obesity have become more prominent in the society. Developing a system that can determine an individual's obesity level without professional intervention can make healthcare be accessible to everyone. Artificial Neural Networks (ANNs) have become more effective lately in determining obesity levels by only asking simple questions regarding lifestyle practices. Using a SMOTE-balanced dataset, this study developed an ANN with an input layer that takes in 16 input features, has two hidden layers activated using the Rectified Linear Unti (ReLU) activation function, and an output layer with a Softmax activation function to classify input into one of the 7 obesity levels. The ANN is evaluated using confusion matrix, accuracy, precision, recall rate, logarithmic loss, and AUC – ROC. The ANN model yielded 90.5% accuracy, 91.9% precision, recall rate of 88.8%, logarithmic loss of 0.3, and AUC – ROC of 0.98.**

1. INTRODUCTION

Obesity, as defined by the World Health Organization, is the excessive or abnormal buildup of fat in a person's body [1]. It is a condition human of all ages have been facing for many years that brings negative consequences to one's overall health. The ongoing global pandemic had majorly taken all the medical resources and human forces and impacted the resource allocation for other conditions. Knowing that there is a new Omicron variant of the Coronavirus as of November 2021 [2] and as observed in [3] that lockdown policies brought by the pandemic caused increased occurrences of obesity especially on groups belonging to the lower socioeconomic class, there is also an urgent need for the combat of obesity. The lockdown implementation and lack of resources – mainly manpower – would make obesity diagnosis difficult.

Obesity diagnosis, on a regular appointment with a doctor, considers the lifestyle of the patient to assess their obesity level. By allowing patients to self-diagnose by answering lifestyle questions, they would be able to monitor their current obesity level without the supervision of a doctor. The predicament of obesity and its diagnosis can be resolved with the use of technology and advanced computing techniques.

Pelachor and de-la-Hoz-Correa, along with de-la-Hoz-Manotas and other researchers from Universidad de la Costa in Colombia, developed an Obesity Level Estimation Software. The data used in the study were from 712 undergraduate participants within the age range of 18 - 25 years old. The study followed the SEMMA data mining methodology and used three different techniques – decision trees, Bayesian networks, and logistic regression – to determine the best way to model the said data set. Of these, the decision trees technique dominated the other techniques based on precision metrics of recall, TP rate and FP rate, with precision garnering 97.4 percent [4].

It is important to note here that [4] focused on developing a model that would present the gathered data. The study was not concerned with the data contents nor its distribution. Another aspect to consider is that the assessment of the precision level of the produced model were not clearly discussed thoroughly, which might affect the validity of the study.

TABLE I

BMI CLASSIFICATION

Classification	**BMI Index Range**
Underweight	Less than 18.5
Normal	18.5 – 24.9
Overweight	25.0 – 29.9
Obesity I	30.0 – 34.9
Obesity II	35.0 – 39.9
Obesity III	More than 40

$$\text{Mass body index} = \frac{weight}{height^2} \tag{1}$$

Motivated by the previous study, Pelachor and de la Hoz Manotas gathered a new set of data having 17 features concerning physical conditions and eating habits of individuals from Mexico, Colombia, and Peru. There were 485 records collected though a web page for 30 days, with ages of participants ranging from 14 to 61. Each record was labelled using the data classification of WHO and Mexican Normativity, as shown in Table I, by the computed body mass index (BMI) for each data record. BMI is the ratio of the weight (in kilograms) to the square of height (in meters) of an individual. However, the distribution of the obesity level classification was unbalanced and thus, synthetic data was generated using the Weka tool and the SMOTE filter to create a balanced data set with 2111 records, as shown in Figure 1. Using Equation 1 as basis, all data were labeled, producing the class variable NObesity, with one of the following

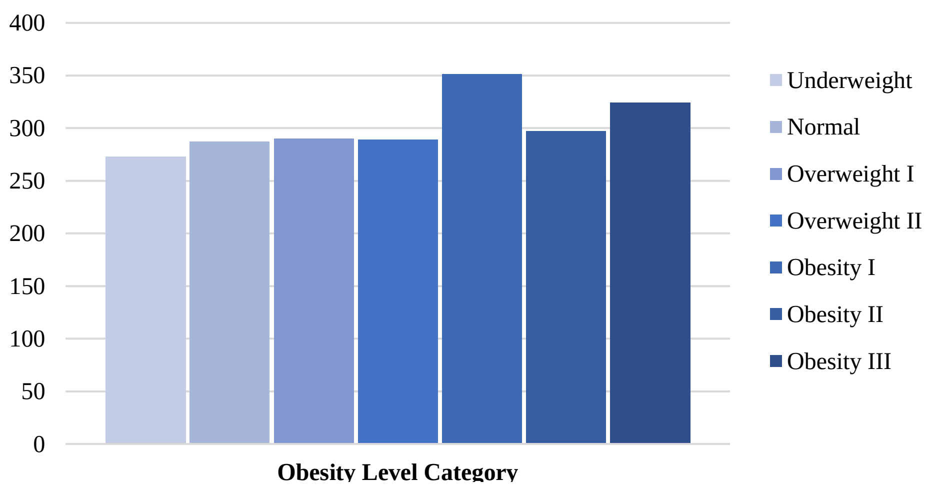

Figure 1. Balanced distribution of dataset for each class.

classifications: Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II, or Obesity Type II [5].

There have been studies that developed neural networks using the balanced data set of Pelachor and de la Hoz Manotas. Kirvak used this data set to produce a deep learning-based prediction of obesity levels with a convolutional neural network, with 82 percent accuracy value, 18 percent classification error value, and 28 percent absolute error value [6]. Garg and Pundir developed the MOFit Framework that predicts obesity level, weight, and body fat percentage. As for the machine learning model for the obesity level prediction, they used a reduced version of the balanced data set – removing the height and weight features – and a random forest model that used grid search and Tpot classifier, which had 86 percent accuracy [7].

This study aims to build an ANN that predicts the Obesity Level given a set of 19 features from the dataset produced by [5] and each set of data would be classified to one of the 7 classes, as labeled by the same study. The performance on the ANN will be analyzed using the metrics of accuracy, logarithmic loss, precision, recall, confusion matrix, and AUC-RUC.

2. METHODS

2.1. *ANN*. The network architecture for this study has an input layer with 19 nodes, 2 hidden layers with 15 and 11 nodes, respectively, and an output layer with 7 nodes. The activation function used by the input layer and the 2 hidden layers is ReLU while the output layer used Softmax. Given that the dataset requires a multiclass classification solution and has many input variables, it would be necessary to use said activation functions.

The ReLU or the Rectifier Linear Unit function is an activation function said to be much better replacement to the Sigmoid function. This function, which has the form

$$ReLU(x) = \max(0, x) = \begin{cases} 0, & if\ x \leq 0 \\ x, & if\ x > 0 \end{cases}$$

is effective on dealing with deeper ANN architectures and solving for gradient problems [8]. Compared to the Sigmoid and Tanh activation functions, ReLU is computationally simpler [9], acts as a linear activation function, have the capacity of have a true value of 0 (Representative Sparsity) [10], and is suitable on dealing with tasks that need Deep Learning.

Softmax function is a multi-class logistic regression that is usually used on the last layer of a given network and that the nodes of the last layer must have the same amount to that of the number of classifications possible [10, 11]. The Softmax function, a mentioned in [12], takes the form of

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}\ for\ j = 1, \dots, K.$$

and is usually used to simultaneously identify the probability of many classes at once since the function of the outputs to [0,1] [10].

The network uses the categorical cross-entropy loss function. It is a loss function that, when applied to categorical data, gives "probabilistic log-likelihood" that can be used for estimating classes and properties [13]. This function, represented by

$$L_{cross-entropy}(\hat{y}, y) = -\sum_{x} y_x \log(\hat{y}_x)$$

measures the difference between the actual label vector y and the predicted label $\hat{y}$ [14].

The network also implemented an Adam optimization. This optimization is "a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments". The default learning rate (α) of the optimization is 0.001, exponential decay rate for first moment estimates (β_1) is 0.9 and for the second moment estimates (β_2) is 0.999 [15].

Training the neural network requires the use of backpropagation algorithm. In this algorithm, initial weights – which are set to random – are loaded to the network, data is processed, weights are shifted to converge to its optimal values, and is repeated until optimum is reached. A single pass through of this algorithm is an epoch. However, in this network, data is processed by batch, which indicates that weights will be shifted after a batch is finished, and there can be multiple batches in an epoch. Along with the consideration of the division of datasets, the number of epochs and the batch size were set to 25 and 20, respectively.

2.2. *Datasets*. The dataset used in this study is the generated dataset by [5], with 2111 entries and 17 attributes. The division of the Training, Validation, and Testing sets is 60 percent, 20, percent, and 20 percent, respectively.

It is important to note that the dataset, as declared in [5], is balanced, with the distribution of the classes shown in Figure 1, and SMOTE filter was used to balance the dataset. The sampling method that is used on the dataset is stratified sampling, and subsets for Training, Validation, and Testing sets were collected on each class and are merged later. The selection in the class for the subsets, as written in the program, is that the first 60% of the class will go the Training set, the following 20% will go to the Validation set, and the remaining 20% will go to the Testing set. Since there are a mix of synthetic and natural data for all classes and data sizes are not distinguished, each class were shuffled five times before the collection of subsets were done to remove possible bias on synthetic or natural data.

2.2.1. *Inputs*

The dataset originally has 16 input features, and each feature – along with its description, original possible values that it can have, and the encoding applied – is shown on Table II. These values were presented in [5].

Of the 16 input features, 14 of those are related to eating habits. The inputs FAVC, SMOKE, and SCC are qualitative binary variables and are encoded using binary representation, with 1 representing Yes and 0 representing No. The inputs CAEC, CALC, and MTRANS are qualitative multiclass variables. However, only the MTRANS variable is encoded using binary representation while the remaining variables were encoded using integer/ordinal representation. The FCVC variable, despite being declared in [5] as a qualitative multiclass variable, is already encoded in the dataset using numeric values.

TABLE II

INPUT FEATURES AND POSSIBLE VALUES

Input Feature	Possible Dataset Values	
	Original Answers	**Equivalent Encoding**
Gender	Female Male	1 0
Age	Numeric Value	
Height	Numeric Value (m)	
Weight	Numeric Value (kg)	
family_history_with_overweight	Yes No	1 0
FAVC (Frequent consumption of high caloric food)	Yes No	1 0
FCVC Frequency of vegetable consumption)	Numerical Value	
NCP (Number of main meals)	Numeric Value	
CAEC (Food consumption freq. in between meals)	No Sometimes Frequently Always	0 1 2 3
SMOKE (Smoking)	Yes No	1 0
CH2O (Daily water intake in liters)	Numeric Value	
SCC (Caloric consumption monitoring)	Yes No	1 0
FAF (Physical activity frequency in days)	Numeric Value	
TUE (Technology use in hours)	Numeric Value	
CALC (Alcohol consumption frequency)	No Sometimes Frequently Always	0 1 2 3
MTRANS[1] (Usually used transportation mode)	Automobile Motorbike Bike Public_Transportation Walking	0000 0100 1000 0010 0001

[1] The assignment of binary values by the program appears to be according to alphabetical arrangement of the possible original values. Since the encoding of MTRANS variable used a reduced version of binary representation to ease the computational demands of the variable, the 'Automobile' value was set to all zeros.

The variables encoded using integer/ordinal representation were encoded by using 0 for No, 1 for Sometimes, 2 for Frequently, and 3 for Always because it is less computationally demanding [16] and the inherent implications of ordinal encoding (average of No and Frequently is Sometimes) is suitable for those variables. Binary representation was applied on the MTRANS using one-hot encoding since the implications of ordinal encoding are not suitable for the data represented by the MTRANS variable. The usage of one-hot encoding on the MTRANS variable translated the insertion of four additional columns on the dataset, with each column header equal to the possible MTRANS variable original values and the value in these columns are set to binary. The assignment of the binary representation in MTRANS variable was selected at random by the program.

The other remaining variables related to eating habits and the age variable are quantitative variables. Given that the dataset was balanced using the SMOTE filter, there are instances where the values for some of these variables are continuous, which must be discrete given the nature of the variable. These occurrences led to all quantitative variables to be encoded using continuous representation.

The remaining Gender variable was encoded using binary representation, with 1 representing Female and 0 representing Male. The justification for this decision is that following the patterns for qualitative binary variables, the Yes option comes first than the No as per [5]. Following that pattern and the data presented in [5] where the Female value for the Gender variable was presented first, the Female value is encoded as 1.

After encoding and splitting the data into Training, Validation and Testing sets, all inputs were standardized using the StandardScaler from the Sci-Kit Learn Preprocessing Library. This process is to standardize features and is executed by “removing the mean and scaling to unit variance” [17].

2.2.2. *Outputs*

Each instance of a group of input variable values is classified into one of the seven obesity levels: Underweight, Normal, Overweight I, Overweight I, Obesity I, Obesity II, and Obesity III. In the seventeenth column in the generated dataset, the Underweight level is indicated with “Insufficient_Weight” while the others are indicated equally to its level name, with spaces replaced with underscores.

Like the MTRANS variable, the obesity levels are encoded using one-hot encoding. However, since this variable is the output for this study and is categorical in nature, the program first has used the LabelEncoder from the Sci-Kit Learn Preprocessing Library. This encoder transforms values of a vector and is used on normalizing labels using integers [18]. Because the target is to encode it using one-hot encoding, the normalized labels were sent through a To Categorical function from the Utils Library of Keras. This function takes in a vector containing classes and returns a binary class sparse matrix [19], which will be needed by the neural network for evaluation and assessment of its performance.

2.3. *Quality Assurance.* For measuring the performance of the neural network in predicting obesity level using input variables, this study has included Accuracy, Confusion Matrix, Precision, Recall, Log Loss, and Area Under the ROC as metrics for our network. The presence of these metrics is under the consideration that the neural network to be examined is a categorical multiclass classifier. The study used the confusion_matrix class to generate the confusion matrices and other metrics were attained using their corresponding classes from the Sci-Kit Metrics library.

2.3.1. *Accuracy.*

As described in [20] and [6], it is the ratio of correct predictions over the total number of predicted samples. While not all studies that used the same dataset used this metric for the assessment of their network/s, Accuracy provides a simple yet significant figure on the capabilities of an established neural network based on the frequency of the network classifying a sample identical to its true class assignment. In formula form, Accuracy is measured by

$$\text{Accuracy} = \frac{\text{\# of Correct predictions}}{\text{Total \# of predictions made}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}}$$

where TP and TF are the number of True Positives and True Negatives, and the FN and FP are the False Positives and False Negatives.

2.3.2. *Confusion Matrix.*

A neural network designed to classify data in multiple classes would best show the complete and detailed performance of the network through a confusion matrix. Present in all previous studies related to this dataset, this metric for performance evaluation of the network was chosen. The confusion matrix is an N-by-N matrix that contains the tally information of TP, TN, FN, and FP for all classes involved in the process. For this study, since there are seven classes in which data can be classified, the size of the confusion matrix is seven.

2.3.3. *Precision.*

This metric is used to identify the proportion of truly positive classifications over all the predicted positive classifications made by the neural network. Observed in [4], this metric is chosen to focus on the predictive performance of the network on positive classifications. The formulaic form for precision is

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

2.3.4. *Recall.*

This metric targets to measure the proportion of successful positive classifications over all the truly positive classifications required to be classified correctly. This value can be attained by

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

2.3.5. *Log Loss.*

Unlike the previously discussed metrics, Log Loss or Logarithmic Loss will assign a probability to each class signifying the chance of a random sample to be classified in that class. If random sample is wrongly classified, a corresponding penalization based on those assigned probabilities will be given. This metric is said to work well on multiclass classifications. This metric, described as

$$\text{LogarithmicLoss} = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

where y_{ij} identifies whether sample belongs to a given class and p_{ij} holds the probability for sample assignment, is important to be minimized to enhance Accuracy [20].

2.3.6. *Area under the ROC (AUC – ROC)*. This metric "quantifies the model's ability to distinguish between each class". The value in this metric is within the range [0, 1] and indicates that the higher the value is, the more capable the network is to differentiate classes, but does not guarantee that said network becomes a better classifier. This metric was selected over the F1 score metric because F1 score metric is used on datasets with large imbalance on classes. Given that our dataset is balanced, the AUC – ROC metric was used [21]. The number of thresholds in discretizing the ROC curve is set to 200 and sample weight is 1 [22].

3. RESULTS AND DISCUSSION

While the model was being trained and validated, accuracy and logarithmic loss were monitored. In Figure 1, the difference of the accuracy and logarithmic loss on the Training Set and Validation Set is only by a small margin, which indicates that the performance of the model on the given datasets is impressive. It is important to note that the significance of the epochs being set to 25 is because increasing it would widen the gap between the two sets. Having a significant difference in accuracy and logarithmic loss between the two sets would imply that the neural network is overfitting.

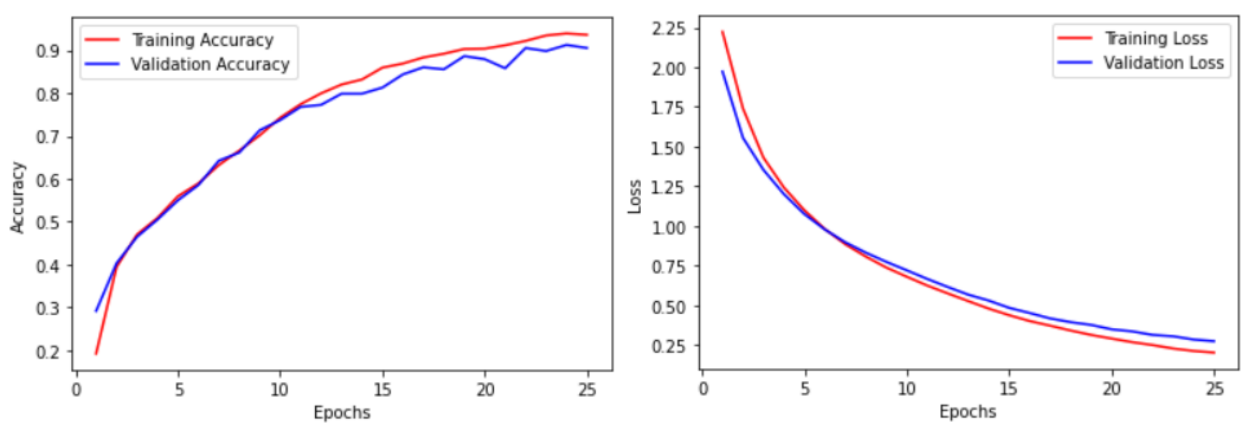

Figure 2. Accuracy development and Loss reduction of the model during ANN training.

After training, the ANN obtained an accuracy of 94.7%, precision of 95.6%, and recall rate of 93.6%. The logarithmic loss yielded only 0.18 and the AUC – ROC was 0.99. The results of the metrics can be observed in the confusion matrix of the ANN to the Training set, in Figure 3.

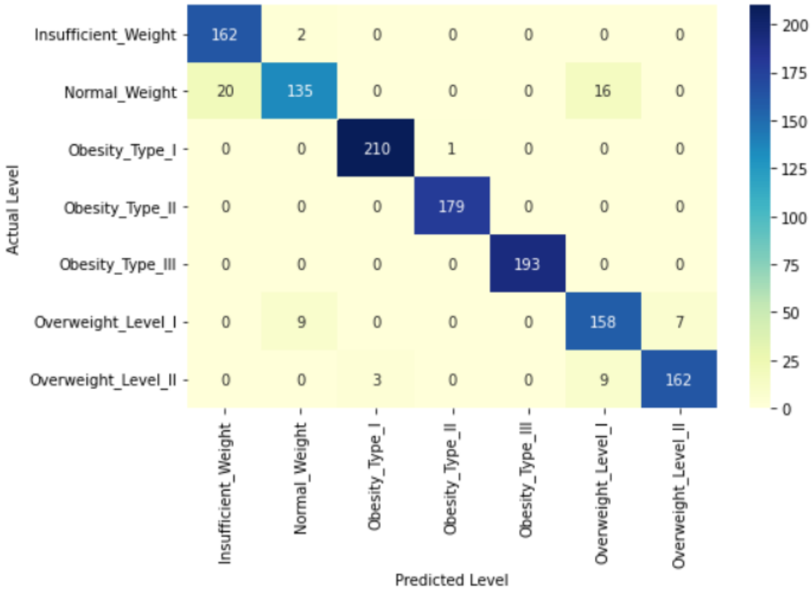

Figure 3. Confusion Matrix of the ANN to the Training Set.

Exposing the ANN to the Testing set, the ANN performed significantly well, with 90.5% accuracy, 91.9% precision, and 88.8% recall rate. The logarithmic loss yielded 0.3 and the AUC – ROC was 0.98. The values seem to be reflected on the confusion matrix of the ANN to the Testing set, in Figure 4.

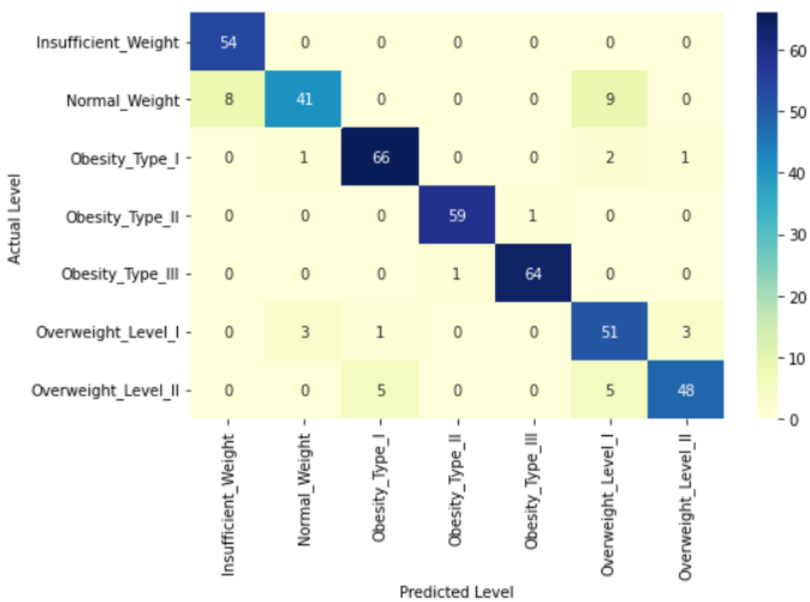

Figure 4. Confusion Matrix of the ANN to the Testing Set.

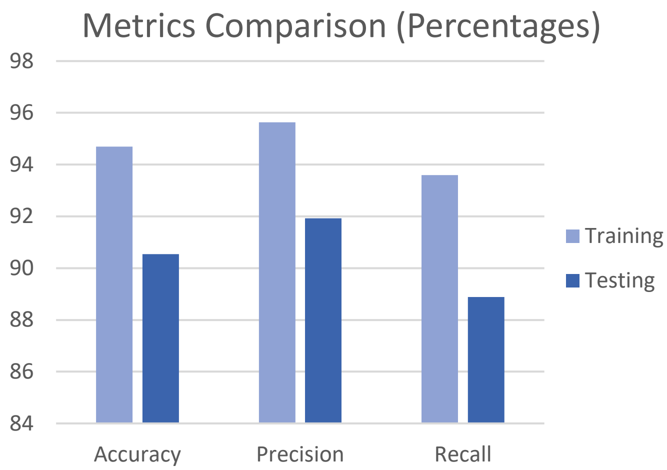

Figure 5. Comparison of Percentage Metrics of ANN Performance

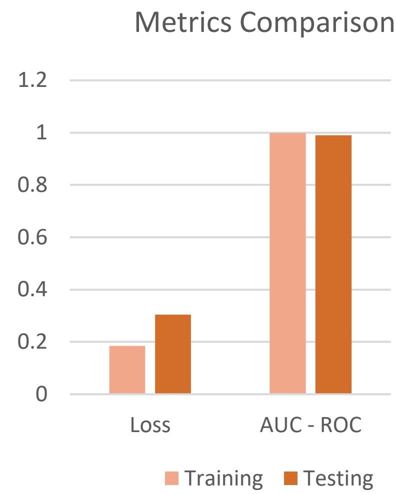

Figure 5. Loss and AUC-ROC Comparison of ANN Performance

Given the accuracy and loss tracking of the model during training, the difference of the ANN results in Training and Testing sets was considerably low on all the metrics that were used to monitor the network's performance, as seen on Figure 5 and 6. The confusion matrices resulted by the network on both sets were very identical in terms of its distribution along the matrix and are very much desirable since the ANN was able to classify the dataset accordingly. The advantage of the dataset as a roughly balanced dataset before even giving it to the network could have probably affected the outstanding performance of the network

4. CONCLUSION

This study has constructed an ANN with the purpose of classifying a set of 16 input features related to eating habits, body measurement, and gender. The ANN was able to classify sets of input features into one of the seven obesity levels and its performance were analyzed through confusion matrix, accuracy, precision, recall, logarithmic loss, and AUC – ROC.

The developed ANN, despite its desirable output, can be analyzed through other metrics to inspect it on other domains, check its integrity, and ensure that there is no overfitting that occurred while training the network. Altering the encoding on variables can also be done to measure whether absolute zero values on input features affect the overall reliability and efficiency of the ANN.

REFERENCES

[1] World Health Organization, "Obesity." [Online]. Available: https://www.who.int/health-topics/obesitytab=tab1
[2] M. Schubert, F. Bertoglio, S. Steinke, P. A. Heine, M. A. Ynga-Durand, F. Zuo, L. Du, J. Korn, M. Milosevic, E. V. Wenzel, H. Maass, F. Krstanovic, S. Polten, M. Pribanic-Matesic, I. Brizic, A. Piralla, F. Baldanti, L. Hammarstrom, S. Dubel, A. Sustic, H. Marcotte, M. Strengert, A. Protic, Q. Pan Hammarstrom, L. Cicin-Sain,

and M. Hust, "Human serum from sars-cov-2 vaccinated and covid-19 patients shows reduced binding to the rbd of sars-cov-2 omicron variant," medRxiv, 2022. [Online]. Available: https://www.medrxiv.org/content/early/2022/01/01/2021.12.10.21267523
[3] C. Clemmensen, M. B. Petersen, and T. I. Sørensen, "Will the covid-19 pandemic worsen the obesity epidemic?" Nature Reviews Endocrinology, vol. 16, no. 9, pp. 469–470, 2020.
[4] E. De-La-Hoz-Correa, F. E. Mendoza-Palechor, A. De-La-Hoz-Manotas, R. C. Morales-Ortega, and S. H. B. Adriana, "Obesity level estimation software based on decision trees," Journal of Computer Science, vol. 15, no. 1, pp. 67–77, Jan. 2019. [Online]. Available: https://thescipub.com/abstract/jcssp.2019.67.77
[5] F. M. Palechor and A. de la Hoz Manotas, "Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from colombia, peru and mexico," Data in Brief, vol. 25, p. 104344, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340919306985
[6] M. Kivrak, "Deep learning-based prediction of obesity levels according to eating habits and physical condition," The Journal of Cognitive Systems, vol. 6, no. 1, pp. 24–27, 2021.
[7] S. Garg and P. Pundir, "Mofit: A framework to reduce obesity using machine learning and iot," 2021.
[8] D. Pedamonti, "Comparison of non-linear activation functions for deep neural networks on mnist classification task," arXiv preprint arXiv:1804.02763, 2018.
[9] J. Feng and S. Lu, "Performance analysis of various activation functions in artificial neural networks," in Journal of physics: conference series, vol. 1237, no. 2. IOP Publishing, 2019, p. 022030.
[10] M. A. Mercioni and S. Holban, "The most used activation functions: Classic versus current," in 2020 International Conference on Development and Application Systems (DAS). IEEE, 2020, pp. 141–145.
[11] Google Developers, "Multi-Class Neural Networks: Softmax." [Online]. Available: https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax
[12] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," towards data science, vol. 6, no. 12, pp. 310–316, 2017.
[13] E. Gordon-Rodriguez, G. Loaiza-Ganem, G. Pleiss, and J. P. Cunningham, "Uses and abuses of the cross-entropy loss: case studies in modern deep learning," 2020.
[14] K. Koidl, "Loss functions in classification tasks," School of Computer Science and Statistic Trinity College, Dublin, 2013.
[15] Keras, "Adam class." [Online]. Available: https://keras.io/api/optimizers/adam/
[16] A. C. H. Choong and N. K. Lee, "Evaluation of convolutionary neural networks modeling of dna sequences using ordinal versus one-hot encoding method," in 2017 International Conference on Computer and Drone Applications (IConDA). IEEE, 2017, pp. 60–65
[17] Scikit learn, "sklearn.preprocessing.StandardScaler." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[18] Scikit learn, "sklearn.preprocessing.LabelEncoder." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html?highlight=label%20encoder#sklearn.preprocessing.LabelEncoder
[19] TensorFlow, "tf.keras.utils.to_categorical." [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/utils/to_categorical
[20] Towards Data Science, "Metrics to Evaluate your Machine Learning Algorithnm." [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234
[21] Towards Data Science, "Comprehensive Guide to Multiclass Classification Metrics." [Online]. Available: https://towardsdatascience.com/comprehensive-guide-on-multiclass-classification-metrics-af94cfb83fbd
[22] Keras, "Classification metrics based on True/False positives & negatives." [Online]. Available: https://keras.io/api/metrics/classification_metrics/#auc-class

APPENDIX I

GitHub Repository of the Study:
https://github.com/johnmichaelt/cmsc_191_paper_neural_network