Enhancing Disease Prediction on Imbalanced Metagenomic Dataset by Cost-Sensitive

Hai Thanh Nguyen[1]
College of Information and
Communication Technology
Can Tho University
Can Tho, Vietnam

Toan Bao Tran[2]
Center of Software Engineering, Duy Tan University,
Da Nang, 550000 Vietnam
Institute of Research and Development, Duy Tan University,
Da Nang, 550000 Vietnam

Quan Minh Bui[3]
College of Information and
Communication Technology
Can Tho University
Can Tho, Vietnam

Huong Hoang Luong[4]
Department of Information Technology
FPT University
Can Tho, Vietnam

Trung Phuoc Le[5]
Department of Information Technology
FPT University
Can Tho, Vietnam

Nghi Cong Tran[6]
National Central University
Taoyuan, Taiwan

***Abstract*—Imbalanced datasets usually appear popularly to many real-world applications and studies. For metagenomic data, we also face the same issue where the number of patients is greater than the number of healthy individuals or vice versa. In this study, we propose a method to handle the imbalanced datasets issues by Cost-sensitive approach. The proposed method is evaluated on an imbalanced metagenomic dataset related to Inflammatory bowel disease to do prediction tasks. Our method reaches a noteworthy improvement on prediction performance with deep learning algorithms including a MultiLayer Perceptron and a Convolutional Neural Neural Network with the proposed cost-sensitive for Metagenome-based Disease Prediction tasks.**

***Keywords*—Cost-sensitive; imbalanced datasets; disease prediction; deep learning**

I. Introduction

The history of medicine is the struggle against the disease based on "one size fits all" strategy. In general, this strategy treats patients who have the same diseases in the same way but in several special cases, that may not be the best treatment for specific patients. Recently, tremendous headway has been proposed in personalized health care, also referred to as precision medicine or personalized medicine. Precision medicine incorporates the insights on environmental, behavioral factors, genome, or biology of a patient.

More specifically, the genetic profiles, and several personal records of the patients are analyzed for identifying the factors of a specific disease, the treatment and prevention can be applied to each patient. It does not only prevent the influence of side effects but also ensures better outcomes. Precision medicine demands to provide the right treatments to the right person at the right time. Furthermore, several studies reveal the contribution of microbiomes on health and disease are considered as a part of precision medicine [1], [2]. The human body contains trillions of bacteria and other microbes and these microbial communities have been examined whole-genome sequencing by the study namely The Human Microbiome Project (HMP) [3]. Metagenomic can be considered as an alternative approach for clarifying the relationship between microbial communities and host phenotype. Furthermore, the discovery of vast new genealogy of microbial life can be developed based in the analysis of 16rRNA sequences from the uncultured microorganisms which represent for the massive majority of creature [4], [5], [6]. Besides, leveraging metagenomic in personalized medicine might take care of many crucial issues [7], [8].

The metagenomic data analysis has created the opportunity for improving the algorithms for specific disease prediction but there are still challenge in computational methods and relatives. The real-world data collection encountered many difficulties and almost the collected datasets are imbalanced. Normally, in the field of metagenomics, the interesting classes have fewer samples than the others and the performance of predicting true label for interesting classes are extremely necessary and important. The cost of a misclassified majority class is usually lower in comparison with the cost of misclassified minority class [9]. The imbalanced ratio affects the performance seriously. Several classic classifiers tried to maximize the validation accuracy and bypass the sensitivity of each class.

II. Related Work

In the field of metagenomic analysis, data pre-processing is truly important and can improve predictive performance. The study [10] proposed a deep learning framework, namely DeepMicro to represent microbiome profiles effectively. Several auto-encoders and machine learning algorithms are used to transform from high-dimensionality of microbiome data into low-dimensional. However, there is still a challenge with meaningful and noisy information. To leverage the meaningful information, the meaningful information should be contained by the learned representation due to encoding of the properties of the input are depended on auto-encoders.

The limitations of data still challenge for several studies in metagenomics fields. The study [11] presented an approach for boosting the performance based on generated metagenomics data. The authors employed a Conditional Generative Adversarial Network (CGAN) to generate the samples which are very similar to the original samples. The predicting host phenotype performance has been improved by augmenting the training dataset. Data augmentation is a common technique to improve performance and generalization in machine learning [12]. Additionally, the authors in [13] also stated the performance of prediction can be able to boost by using Generative Adversarial Network (GAN) models. Nevertheless, selecting the best CGAN model is still a difficult task, the optimal model can be bypassed.

The study [14] presented a machine learning approach for diagnostic decisions. The predictive model is simple but gains a powerful score by computing the cumulative abundance of microbiome measurements. However, the performance of predictive models can be affected by data quantification problems. More specifically, the individuals and specific types of microbial ecosystems have a significant difference in microbial loads [15]. Furthermore, the model can select various sequencing depth can be over or underestimate less abundant taxa.

In this study, we propose a Cost-Sensitive method [16], [17] to handle the imbalanced datasets issue. Thereby, enhancing the performance on disease prediction task. Our contributions include the following:

- We present the considered datasets and handling imbalanced issues with the Cost-Sensitive method.
- The efficiency of the proposed methods is evaluated on three types of learning models including Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN). The performance with the Cost-Sensitive method obtains better results for each learning model.

In the remaining of this study, we introduce the considered dataset in Section III. The learning algorithms and Cost-Sensitive are proposed in Section IV. Section V presents the compared performance of each learning model in cases of before and after applying the Cost-Sensitive method. We discuss and summarize the results in Section VI.

III. Imbalanced Datasets in Metagenomics

The proposed approach performance is evaluated on the Inflammatory Bowel Disease (IBD) dataset [18], more details are in Table I. The details of the considered datasets including the numbers of features, samples, patients, and several additional information.

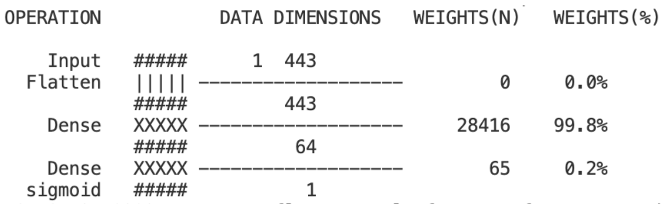
Table I indicates that the number of samples and patients is widely large, it is a basic case of imbalanced datasets. Imbalanced datasets are relevant primarily in the context of a classification task where the class distribution is not uniform among the classes. The considered dataset contains 25 patients and 110 samples, the patient ratio of 0.23, and the ratio of the control of 0.77. Each sample or patient includes 443 discriminate features. Each feature reflects the proportion of a bacterial species existing in a sample's body.

The total value of all features (relative abundance of bacterial species) in one patient or a healthy individual is sum up to 1 (as shown in Equation 1):

$$\sum_{i=1}^{k} f_i = 1 \qquad (1)$$

TABLE I. Details of Inflammatory Bowel Disease (IBD) Dataset.

Dataset	**IBD**
Features	443
Samples	110
Patients	25
Controls	85
Ratio of patients	0.23
Ratio of controls	0.77



```
OPERATION           DATA DIMENSIONS   WEIGHTS(N)   WEIGHTS(%)

   Input   #####      1  443
 Flatten   |||||  -------------------         0     0.0%
           #####        443
   Dense   XXXXX  -------------------     28416    99.8%
           #####         64
   Dense   XXXXX  -------------------        65     0.2%
 sigmoid   #####          1
```

Fig. 1. Visualization of the Multilayer Perceptron Architecture used in the Experiments.

With:

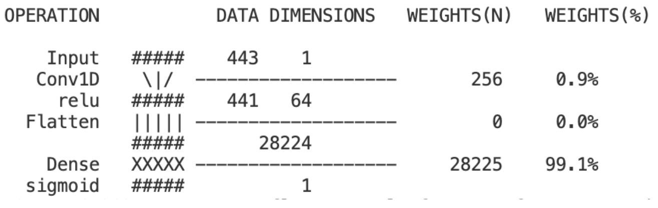
- k is the number of features for a sample.
- f_i is the value of the i-th feature.

IV. Cost-Sensitive Approaches in Deep Learning Algorithms for Imbalanced Datasets

A. Cost-Sensitive Methods

The goal of cost-sensitive learning for imbalanced classification tasks is to assign different costs to misclassification errors and compute those costs by specialized methods. A confusion matrix is a powerful tool for summarizing the predictions for the individuals and shows how well a method performs on a prediction. It allows the visualization of the performance of a learning algorithm. There are several common cost-sensitive methods including Cost-Sensitive Resampling, Cost-Sensitive algorithms, or Cost-Sensitive Ensembles. In this study, we investigate the performance of Cost-Sensitive Algorithms on an imbalanced metagenomic dataset.

The training section of learning algorithms uses the back-propagation to compute the error on the training set and update the weights based on those errors. However, the samples of each class are trained the same as each other, in the case of the imbalanced dataset, the model focus on the majority class more than the minority class. During back-propagation, the weight of misclassification errors can be updated in proportion to the



```
OPERATION           DATA DIMENSIONS   WEIGHTS(N)   WEIGHTS(%)

   Input   #####    443    1
  Conv1D    \|/  -------------------       256     0.9%
    relu   #####    441   64
 Flatten   |||||  -------------------         0     0.0%
           #####      28224
   Dense   XXXXX  -------------------     28225    99.1%
 sigmoid   #####          1
```

Fig. 2. Visualization of the Convolutional Neural Network Architecture used in the Experiments.

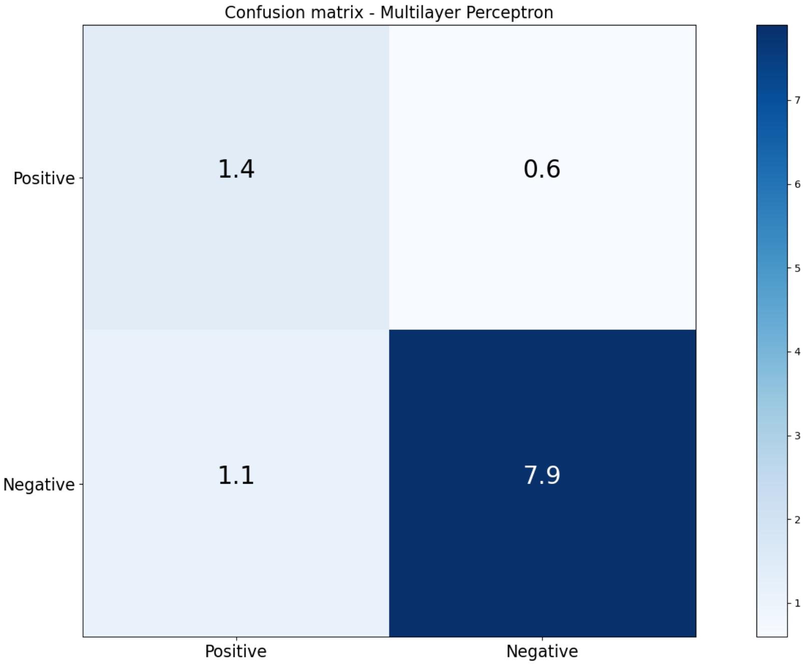


Fig. 3. The Average Confusion Matrix Result Running by Multilayer Perceptron with Cost-Sensitive Method.

importance of the class and have an influence on the model to pay more attention to samples of the minority class.

In our experimental results, computing the class weight from class distribution present in the training dataset can improve the performance effectively. The class weight can be computed by the Equation 2 inspired by the study [16].

$$w_c = \frac{n}{t * s_c} \tag{2}$$

Where w_c denotes the weight of class c, n represents for the number of samples in training set, t is number of class and s_c stands for the number of samples in class c.

B. Learning Models

To investigate the performance of training with Cost-Sensitive method and training without Cost-Sensitive methods, we used different architectures as mentioned above.

The Multilayer Perceptron (MLP) is a class of Artificial Neural Network (ANN), a MLP contains at least tree node layers. The input layer aims to receive the data while the hidden layers are the primary computational engines. The output layer produces the prediction result. The architecture of details of MLP used in the experiments is presented in Fig. 1.

Finally, the Convolutional Neural Networks (CNN) contains a $1D$ Convolutional layer, followed by a Fully Connected layer. The model learns an internal representation of a two-dimensional input, in a process referred to as feature learning. We visualized the CNN architecture in Fig. 2.

All three learning models are implemented with Adam optimizer [19], the default learning rate is 0.001. The Early Stopping method is also applied to avoid overfitting issues with a patience epoch of 5.

C. Metrics for Comparison

To evaluate the classification performance, we used three metrics namely Accuracy (ACC) and Area Under the Receiver Operating Characteristic Curve (ROC-AUC), and Matthews correlation coefficient (MCC). We investigated the accuracy and AUC of training with cost-sensitive and training with non-cost-sensitive. The accuracy and MCC are computed by the Equation 3 and Equation 4 respectively.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4}$$

Where

- TP denotes True Positive.
- TN denotes True False.
- FP denotes False Positive.
- FN denotes False False.

Furthermore, the MCC is, in essence, a correlation coefficient of binary classifications. The MCC value has a range of -1 to $+1$. A coefficient of $+1$ represents a completely correct binary classifier, 0 stands for random prediction, whereas -1

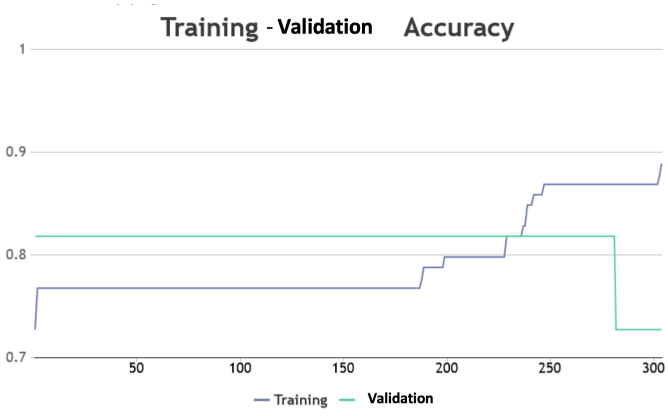


Fig. 4. Visualization of Training and Validation Accuracy of Multilayer Perceptron. X-axis Shows the Number of Epochs used in Training Phase while Y-axis Reveals Accuracy.

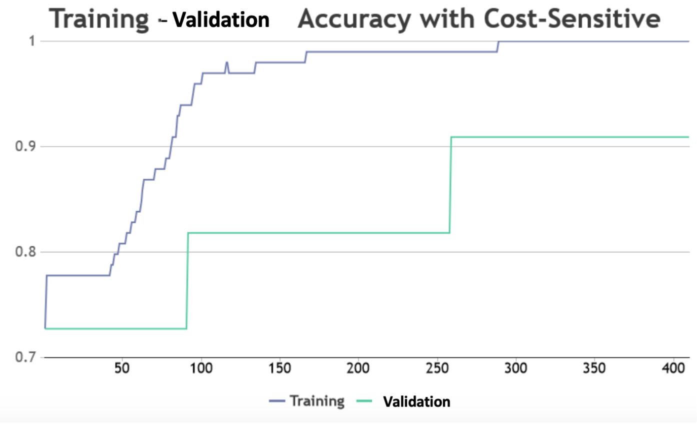


Fig. 5. An Illustration of Training and Validation Performance in Accuracy with Multilayer Perceptron Model using Cost-Sensitive Method. X-axis Shows the Number of Epochs used in Training Phase while Y-axis Reveals Accuracy.

indicates total disagreement between prediction and observation.

Another metric is Loss which is also considered. The loss function implemented in networks in the study is Binary Cross-Entropy (Equation 5) [20]. The binary entropy function, denoted $H_p(q)$:

$$H_p(q) = -\frac{1}{N}\sum Ni = 1y_i.log_2(p(y_i))+(1-y_i).log_2(1-p(y_i)) \tag{5}$$

where y is the ground truth and p(y) is the predicted probability of the predicted sample.

V. Experimental Results

We trained all considered deep learning architectures with 10-folds stratified-cross validation. The performance of each model is measured by Accuracy, Area Under Curve (AUC), Matthews correlation coefficient (MCC), and Loss presented as follows.

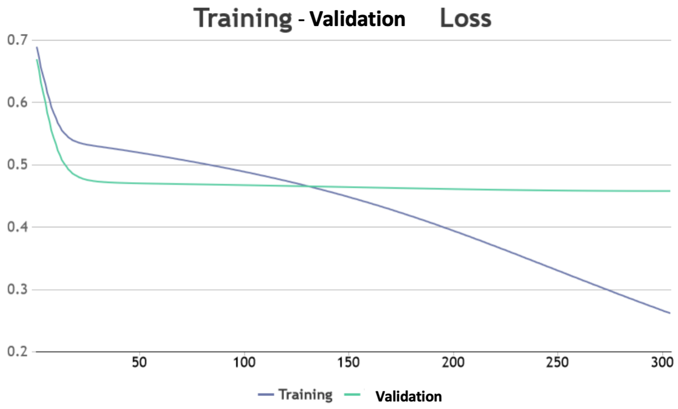


Fig. 6. Visualization of Training and Validation Loss of Multilayer Perceptron. X-axis Shows the Number of Epochs used in Training Phase while Y-axis Reveals Loss.

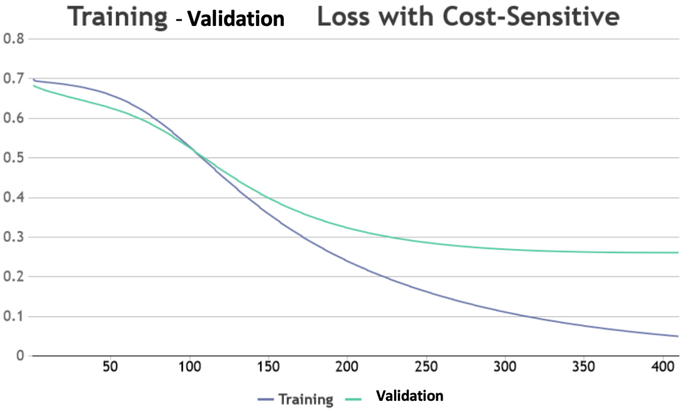


Fig. 7. Visualization of Training and Validation Loss of Multilayer Perceptron with Cost-Sensitive Method. X-axis Shows the Number of Epochs used in Training Phase while Y-axis Reveals Loss.

A. Performance of Multilayer Perceptron

After 10-folds, the MLP obtained 0.77 of average overall accuracy, 0.643 of AUC, and 0.052 of MCC. The results are unsatisfied with the classification task due to the imbalanced dataset. We conducted training the model again with a cost-sensitive method and gained better results. More specifically, the accuracy increased to 0.845, 0.865 for AUC, and 0.552 for MCC. The exceptional increase of MCC stated the model was much better than before. Fig. 3 visualizes the confusion matrix of this learning model with cost-sensitive method. The average True Positive obtained 1.4, False Positive of 0.6, False Negative, and True Negative gained 1.1 and 7.9 respectively.

The training and validation accuracy of the compared methods are visualized in Fig. 4 and Fig. 5. Fig. 4 represents for the performance of non cost-sensitive. Otherwise, Fig. 5 visualizes the results of the cost-sensitive method, the training and validation accuracy got better epoch by epoch with the cost-sensitive method. Also with the loss, Fig. 6 and Fig. 7 present the training and validation loss of the learning model. As observed, the validation loss in Fig. 7 is better in comparison with the other in Fig. 6.

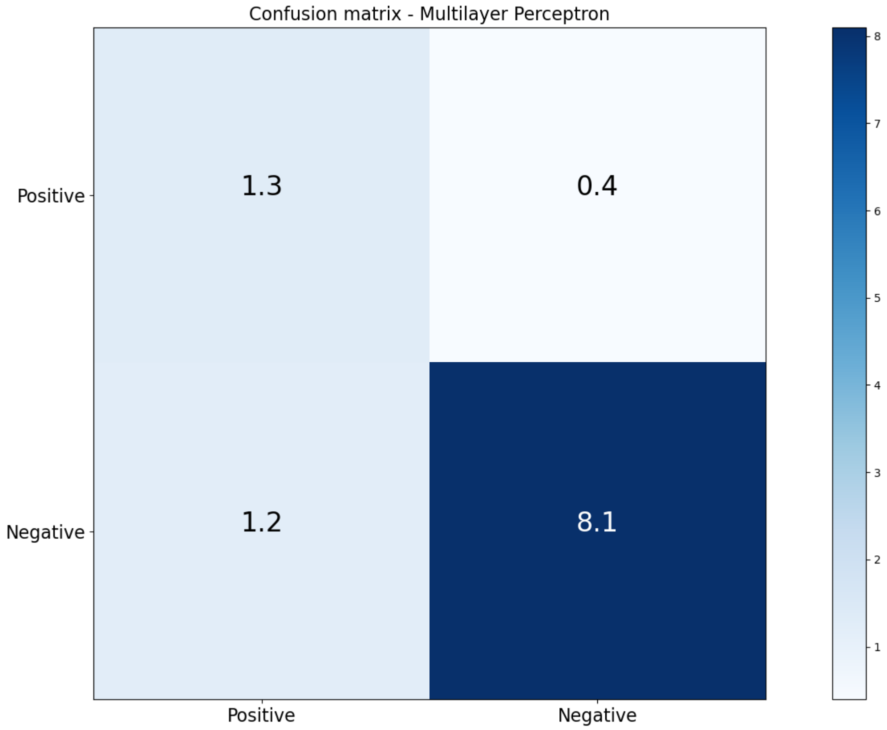


Fig. 8. The Average Confusion Matrix Result Running by on Convolutional Neural Network with Cost-Sensitive Method.

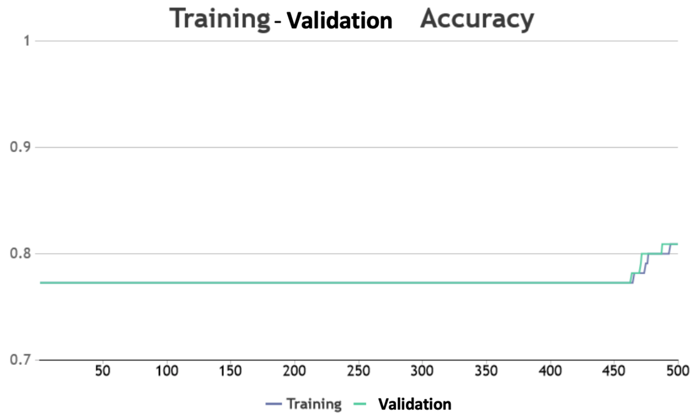


Fig. 9. Visualization of Training and Validation Accuracy of Convolutional Neural Network. X-axis Shows the Number of Epochs used in Training Phase while Y-axis Reveals Accuracy.

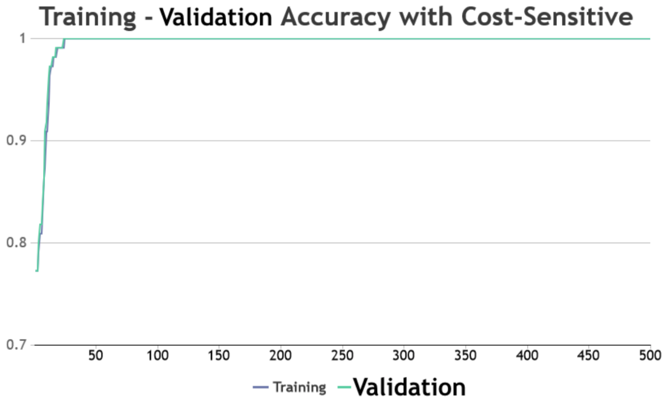


Fig. 10. Results of Training and Validation Accuracy with Convolutional Neural Network Combining Cost-Sensitive Method. X-axis Shows the Number of Epochs used in Training Phase while Y-axis Reveals Accuracy.

B. Performance of Convolutional Neural Network

We also investigated the performance of the Convolutional Neural Network on 10-folds cross-validation. In comparison with Multilayer Perceptron, the performance of the Convolutional Neural Network is very close. The average accuracy reached 0.773, AUC of 0.629, and MMC of 0. The boosted performance with the cost-sensitive method is slightly better, the accuracy increased to 0.855 but the AUC reached 0.871 and the MCC obtained 0.513. By applying the boosting performance method, the AUC of Convolutional Neural Network is better than Multilayer Perceptron whereas the accuracy and MCC are similar.

The confusion matrix of Convolutional Neural Network is visualized in Fig. 8. In comparison with the prior learning model, the values of True Positive, False Positive, False Negative, and True Negative are relatively similar. We also presented the training and accuracy/loss validation of Convolutional Neural Network in Fig. 9, Fig. 10, and Fig. 11, Fig. 12 respectively.

In the comparison of validation accuracy in Fig. 9 and Fig. 10, the validation accuracy of non-cost-sensitive method

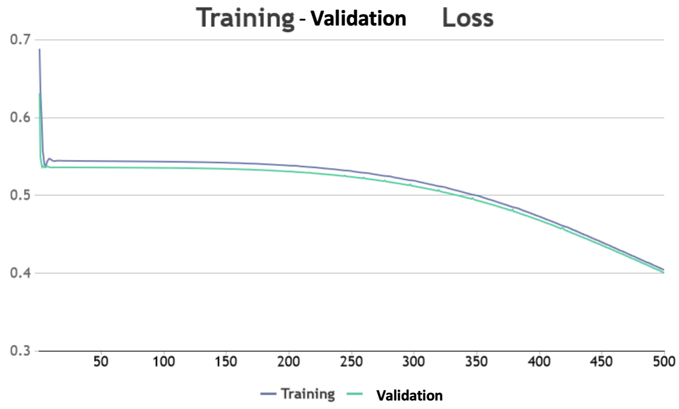


Fig. 11. Visualization of Training and Validation Loss of Convolutional Neural Network. X-axis Shows the Number of Epochs used in Training Phase while Y-axis Reveals Loss.

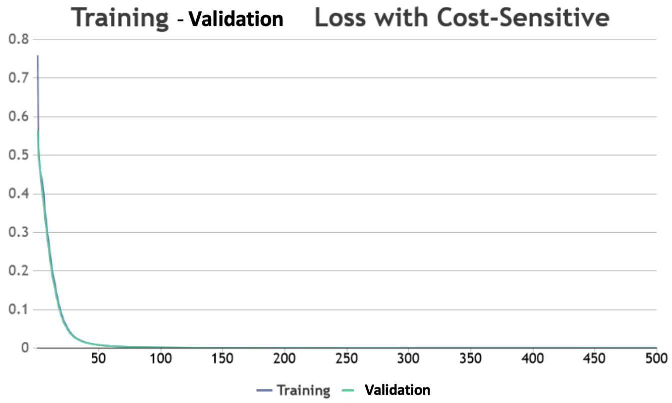


Fig. 12. Visualization of Training and Validation Loss of Convolutional Neural Network with Cost-Sensitive Method. X-axis Shows the Number of Epochs used in Training Phase while Y-axis Reveals Loss.

kept stable at 0.77 in almost training section and peaked at last epochs whereas the accuracy of model applying cost-sensitive reached the optimal performance around 50 epochs. Similar to Loss validation, the loss of non-cost-sensitive method stopped at 0.4 whereas the other is almost equal to 0.

C. Comparison of Multilayer Perceptron and Convolutional Neural Network

We summarized the performance of Multilayer Perceptron and Convolutional Neural Network in Table II. In general, the performance of the two learning models is similar. With the Cost-Sensitive method, the overall accuracy improved slightly but AUC and MCC were significant. The cost-sensitive method affected effectively to the classification performance.

VI. Conclusion

We introduced a method based on a Cost-sensitive approach to improving the performance of imbalanced datasets. The proposed method is efficient on not only Multi-Layer Perceptron but also Convolutional Neural Network.

TABLE II. The Comparison of Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN).

Model	**Accuracy**	**AUC**	**MCC**
MLP	0.770	0.643	0.052
MLP with Cost-Sensitive	0.845	0.865	0.552
CNN	0.773	0.629	0.000
CNN with Cost-Sensitive	0.855	0.871	0.513

The performance is assessed by various metrics including Accuracy, AUC, MCC which reveal significant improvements with the cost-sensitive method. Besides, the proposed method enables the learning model to learn faster as well as speed up the convergence of models.

Further research can investigate more data and test on sophisticated machine learning algorithms.

References

[1] Petrosino, J.F. "The microbiome in precision medicine: the way forward". Genome Med 10, 12, 2018. https://doi.org/10.1186/s13073-018-0525-6

[2] Gilbert JA, Quinn RA, Debelius J, et al. Microbiome-wide association studies link dynamic microbial consortia to disease. Nature. 2016;535(7610):94-103. doi:10.1038/nature18850

[3] Turnbaugh, P., Ley, R., Hamady, M. et al. The Human Microbiome Project. Nature 449, 804–810 (2007). https://doi.org/10.1038/nature06244

[4] Jang SJ, Ho PT, Jun SY, Kim D, Won YJ. Dataset supporting description of the new mussel species of genus Gigantidas (Bivalvia: Mytilidae) and metagenomic data of bacterial community in the host mussel gill tissue. Data Brief. 2020;30:105651. Published 2020 Apr 29. doi:10.1016/j.dib.2020.105651. 2020

[5] Ma, Bing & France, Michael & Ravel, Jacques. (2020). Meta-Pangenome: At the Crossroad of Pangenomics and Metagenomics. doi10.1007/978-3-030-38281-0_9.

[6] Handelsman J. Metagenomics: application of genomics to uncultured microorganisms. Microbiol Mol Biol Rev. 2004;68(4):669-685. doi:10.1128/MMBR.68.4.669-685.2004

[7] Hongyu Chen, Sanjeev Kumar Awasthi, Tao Liu, Zengqiang Zhang. Mukesh Kumar Awasthi, "An assessment of the functional enzymes and corresponding genes in chicken manure and wheat straw composted with addition of clay via meta-genomic analysis", Industrial Crops and Products, vol. 153, 2020, doi:https://doi.org/10.1016/j.indcrop.2020.112573

[8] Alfredo D. Guerron et al. "Performance and Improvement of the DiaRem Score in Diabetes Remission Prediction - A Study with Diverse Procedure Types", May. 2020, doi:https://doi.org/10.1016/j.soard.2020.05.010. 2020.

[9] Chawla, Nitesh & Bowyer, Kevin & Hall, Lawrence & Kegelmeyer, W.. (2002). SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. (JAIR). 16. 321-357. doi:10.1613/jair.953.

[10] Oh, M., Zhang, L. DeepMicro: deep representation learning for disease prediction based on microbiome data. Sci Rep 10, 6026 (2020). https://doi.org/10.1038/s41598-020-63159-5

[11] Reiman, Derek and Dai, Yang, "Using Conditional Generative Adversarial Networks to Boost the Performance of Machine Learning in Microbiome Datasets," bioXiv:2020.05.18.102814, https://doi.org/10.1101/2020.05.18.102814, May 2020.

[12] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," 2018 International Interdisciplinary PhD Workshop (IIPhDW), Swinoujście, 2018, pp. 117-122, doi: 10.1109/IIPHDW.2018.8388338.

[13] Che, Z., Cheng, Y., Zhai, S., Sun, Z., & Liu, Y. (2017). Boosting Deep Learning Risk Prediction with Generative Adversarial Networks for Electronic Health Records. 2017 IEEE International Conference on Data Mining (ICDM), 787-792.

[14] Edi Prifti, Yann Chevaleyre, Blaise Hanczar, Eugeni Belda, Antoine Danchin, Karine Clément, Jean-Daniel Zucker, Interpretable and accurate prediction models for metagenomics data, GigaScience, Volume 9, Issue 3, March 2020, giaa010, https://doi.org/10.1093/gigascience/giaa010

[15] Vandeputte D, Kathagen G, D'hoe K, et al. Quantitative microbiome profiling links gut community variation to microbial load. Nature. 2017;551(7681):507-511. doi:10.1038/nature24460

[16] King, Gary & Zeng, Langche. (2002). Logistic Regression in Rare Events Data. Political Analysis. 9. 10.1093/oxfordjournals.pan.a004868.

[17] Kukar, M., & Kononenko, I. (1998). Cost-Sensitive Learning with Neural Networks. ECAI.

[18] E. Pasolli, D. T. Truong, F. Malik, L. Waldron & N. Segata; Machine Learning Meta-analysis of Large Metagenomic Datasets: Tools and Biological Insights; PLoS Comput. Biol. 12, p. e1004977. 2016.

[19] Diederik P. Kingma and Jimmy Lei Ba. Adam : A method for stochastic optimization. 2014. arXiv:1412.6980v9

[20] Zhang, Zhilu and Sabuncu, Mert. (2018). Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. https://papers.nips.cc/paper/8094-generalized-cross-entropy-loss-for-training-deep-neural-networks-with-noisy-labels.pdf. 2018.