Comparison of Fine-tuned CNN Architectures for COVID-19 Infection Diagnosis

Jonathan[1], Moeljono Widjaja[2], Alethea Suryadibrata[3]
Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia
[1]jonathan8@student.umn.ac.id, [2] moeljono.widjaja@umn.ac.id, [3]alethea.suryadibrata@lecturer.umn.ac.id

Accepted 2 June 2024
Approved 26 June 2024

***Abstract*— SARS-CoV-2 (COVID-19) virus spread quickly worldwide affects a variety of industries. The government took preventive steps to control the infection, such as diagnosing the human's lung by taking an X-Ray to see if the lungs were infected with COVID-19 or not. Using several pre-trained Convolutional Neural Network models as the basic model, this research deconstructs the comparison of fine-tuned architecture to identify which pre-trained model delivers the best outcomes in diagnosis by applying machine learning. Comparison is conducted using two scenarios that use batch sizes 64 and 32. Accuracy and f1 score are two evaluation metrics used to justify the model's good performance because the images in the real world, especially for positive classes, are scarce. According to the study, EfficientNetB0 outperforms other pre-trained models, namely ResNet50V2 and Xception, which achieved an accuracy of 0.895 and f1 score of 0.8871.**

***Index Terms*— Convolutional Neural Network; COVID-19; Machine Learning; X-Ray**

I. Introduction

Because of the rapid spread of the SARS-CoV-2 (COVID19) virus worldwide, many people have been contracted, isolated, and removed from the rest of society to prevent human-~~—~~to-human transmission. Some of the economic consequences, such as economic loss due to flight cancellations (US$245 million) until a significant downfall in hotel occupancy rate (plunged 32.6% compared to June 2020 with June 2019), show how bad the virus affects the world [1]. It also affects the wellbeing of humans itself where the research shows that 41% of respondents feels that their happiness was deteriorated [2]. This caused a pandemic that brought unforeseen crises, including those for creative industry workers [3]. Not only did it affect creative industry workers, but also affected regency politics as explained by Daniel Susilo in his research of fighting the pandemic of COVID-19 in regency [4].Several measures, such as recognizing people exposed to the SARS-CoV-2 virus, were used to control the infection's rapid spread. Detection becomes critical to stop the spread and take preventative measures [5].

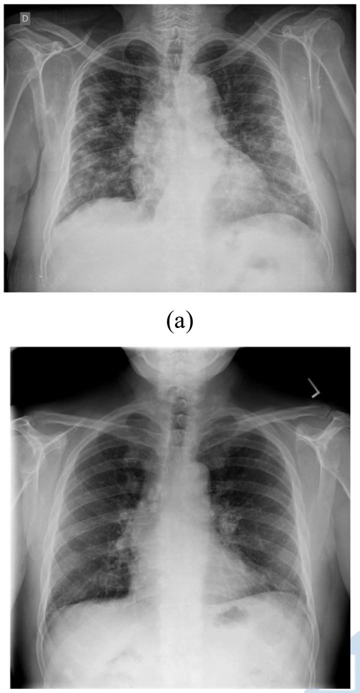
According to WHO [6], the primary symptom of the SARSCoV-2 virus, also known as COVID-19, is respiratory problems. Because respiratory issues indicate that the virus has infected the lungs, the X-Ray image will be able to reveal if the virus is present or not [7~~5~~]. The use of X-Ray scanning machines in hospitals and laboratories can be used to discover this disease as early as possible. On the other hand, the diagnosis procedure is usually done manually by a doctor without technical improvements. Developing a model using a pre-trained model that provides results for detecting whether a person is infected with COVID-19 or not is very promising because it can reduce the time for the doctor to diagnose a patient, which is currently done manually.

Previous studies have shown that Convolutional Neural Network (CNN) is an appropriate method to classify a digital image. In [8], CNN is used to diagnose diabetic retinopathy from fundus image and obtained a model with ROC AUC score of 98%. In [9], CNN is used to classify between bacterial pneumonia and viral pneumonia on chest X-ray dataset.

This research aims to identify the most accurate pre-trained model among ResNet50V2 [10], Xception[11], and EfficientNetB0 [12] for diagnosing COVID-19 infections using the COVIDx CXR-2 chest X-ray dataset. These models consist of a range of architectures, from the old model like ResNet50V2 to the state-of-the-art model like EfficientNetB0. In contrast, many related studies compare older models such as VGG, Xception, ResNet, and Inception. This research also applies some basic fine-tuning by adjusting the batch size from 32 to 64. By determining the best pre-trained model, it can be utilized to train with a larger dataset, ensuring that the model performs well outside of the trained image.

II. Research Method

X-Ray images of human lungs are the subject of investigation in this study. The image will be used to determine whether the lung is infected with COVID-19 or not. Because of its availability and affordable cost, X-Ray is usually the first diagnostic test in patients with suspected or confirmed COVID-19, rather than the RT-PCR test [13]. Figure 1 displays the lungs image of individuals infected with COVID-19 and those who are not infected with COVID-19 for comparison.



(a)

(b)

Fig. 1. Example of a figure caption Human's lung, (a) Positive COVID-19, (b) Negative COVID-19. Source: COVIDx-CXR 2 dataset [10]

Figure 1a shows hazy or cloudy areas around the lungs, which may indicate a COVID-19 infection. In contrast, Figure 1b shows a clear lung X-ray, with no hazy or cloudy areas. This indicate the lung is negative COVID-19.

The dataset used as the study's object was derived from the COVIDx CXR-2 X-Ray lung dataset prepared by Alexander Wong and Linda Wang [14]. Because the dataset's negative and positive samples are unbalanced, they must be pre-processed before being used. Table I shows the properties of the dataset.

TABLE I. DATASET ATTRIBUTES

Data Type	Class Type	Before Preprocessing	After Preprocessing
2*training	Negative	13793	2158
	Positive	2158	2158
2*test	Negative	200	200
	Positive	200	200

To compare the diagnosis result, the researcher trained the model in three pre-trained Convolutional Neural Network (CNN) architectures, namely ResNet50V2, Xception, and EfficientNet B0. The steps taken in each of these architectures are shown in Figure 2.

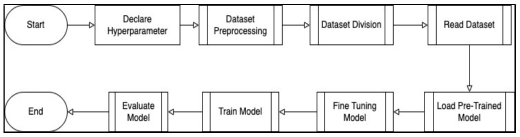


Fig. 2. The Flow of the Research

A. Pre-Trained Model

There are various reasons to utilize a pre-trained model, according to Krishna *et al.* [15]. First, training large models on large datasets requires significant computer power. By utilizing the pre-trained model, it can help reduce the computational burden. Second, pre-trained model can lead to faster outcomes when being fine-tuned. It is possible to reduce training time by using pre-trained models and training new models based on pre-trained weight. As a result, using the pre-trained model as a base model, then training the model with datasets and applying some additional layers is a smart move. The following are the pre-trained models that were used in this study:

1) ResNet50

ResNet consists of a "Residual Unit" stack. ResNet itself introduces a feature that can ignore one or more layers called "skip connections" and are the central part of the residual block. This method solves the problem where if the built network gets more profound, the accuracy will stagnate or not develop.

Gunraj *et al.* [14] use this architecture to diagnose COVID-19 infection. In his journal, the model is trained by three steps where at each stage, the machine will be trained with a different dataset. The training process with different datasets resulted in better accuracy and the ability to detect and ignore the noise in the image [16].

The ResNet architecture employed in this work is ResNet50V2, the second-generation one. Using a preactivation layer before being added to the residual block distinguishes this generation from the first one. In comparison to the first generation [17], the addition of this feature produces promising results.

2) Xception

Xception was developed by Francois Chollet, a Google employee, by modifying depth wise separable convolution. Xception is claimed to outperform the Inception V3 architecture, which became its predecessor in the ImageNet dataset. The number of parameters used is the same as Inception V3. The main difference is that Xception could achieve fewer model parameters and still maintain the results [11].

From Figure 3, it can be seen that the pointwise convolution (1 x 1 Convolution) is performed to change the dimension into a new one before the depth wise convolution (n x n Spatial Convolution). Thus, the model can extract more features in one step.

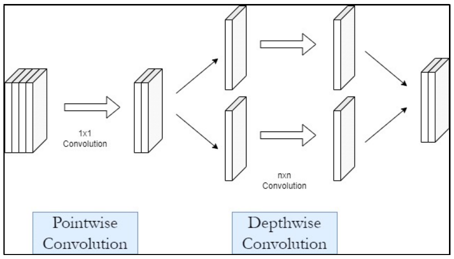


Fig. 3. Modified Depthwise Separable Convolution in Xception

3) EfficientNet

EfficientNet was published by Google in 2019. This model not only increases accuracy but also improves model accuracy by reducing parameters and Floating Point Operations Per Second (FLOPS) as was done in the GPipe architecture where GPipe itself uses Pipeline Parallelism [12].

EfficientNet scales uniformly from the width, depth, and resolution aspects with a fixed coefficient defined as the ϕ symbol. This method is called Compound Scaling. The formula can be written as follows:

$$depth : \delta = \alpha^{\phi} \quad (1)$$

$$width : \omega = \beta^{\phi} \quad (2)$$

$$resolution : \tau = \gamma^{\phi} \quad (3)$$

These fixed coefficients are the coefficients that control the computing resources. For example, if we want to use 2^{ϕ} more computing resources, then we can increase the network depth by α^{ϕ}, the width by β^{ϕ}, and the image size by γ^{ϕ}. The values are constant coefficients determined by tracing the small tile in the original mini model.

In this study, the evaluation metrics that will be used to compare between models are accuracy. Accuracy is the most popular, which usually be the first metric to be calculated in all classification problems. It tells the ratio of accurately classified data items to the total number of observations based on Formula 4 [18].

Using accuracy as the only evaluation criteria is not a good idea, especially when there is a scarcity of data in the actual world and the data is likely to be imbalanced. The nature of accuracy, which implies equal relevance of classes in terms of the number of instances and the level of importance, requires the calculation of the F1 score evaluation metric to account for these concerns [19]. The F1 Score indicates whether the results are biased in favor of the positive or negative class [20].

B. Callback

An overfitting model is not helpful in machine learning research. Overfitting occurs when a learning model is overly focused on the training data, resulting in poor performance when evaluating new data that has not been previously assessed [21]. Several callbacks are implemented at the end of each epoch to avoid constructing the overfitting model, namely:

1) Early Stopping

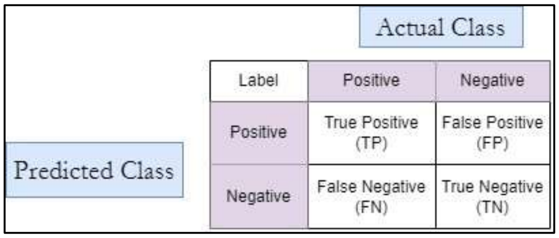
Early Stopping is a callback that stops the model training process when the current model provides the same outcome as the smaller model [22]. As a result, training time can be reduced while ensuring that the model does not become overfit.

2) Reduce Learning Rate on Plateau

Machine learning models use the learning rate to determine how much the weights can change when the training data is evaluated incorrectly. When the learning rate is too high, the model cannot adjust its weight when new data is provided. As a result, as the model plateaus, it is necessary to reduce the learning rate [23].

C. Confusion Matrix

Confusion matrix is a performance measurement of the classification task for machine learning [24]. True Positive (TP) tells that the prediction result gives a positive value and the actual result is positive. This means that the prediction results are correct. False Positive (FP) tells that the predicted result gives a positive value and the actual result is negative. This means that the prediction result is false. True Negative (TN) tells that the predicted result gives a negative value and the actual result is negative. This means that the prediction results are correct. False Negative (FN) tells that the predicted result gives a negative value and the actual result is positive. This means that the prediction result is false. The confusion matrix itself could be presented in the form of a table with a combination of predicted and actual results as shown in Figure 4.



		Actual Class	
	Label	Positive	Negative
Predicted Class	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Fig. 4. Confusion Matrix Table

From the value generated by the confusion matrix, the accuracy value can be calculated. The accuracy formula is as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

D. Classification Report

The classification report has four main values that will be displayed when used. Indirectly, this value requires values from *confusion matrix*. The four main values in the classification report are:

1) *Precision*

The precision value describes the model's ability not to label positive images that are negative. This means a value representing the accuracy of all positive predictive results obtained. This value is calculated by the value of True Positive divided by the number of True Positive by False Positive. The formula is as follows:

$$Precision = TP/(TP + FP) \quad (5)$$

2) *Recall*

The recall value describes the model's ability to find all positive images. This means a value representing the accuracy of all positive image results obtained. This value is calculated by the value of True Positive divided by the number of True Positive by False Negative. The formula is as follows:

$$Recall = TP/(TP + FN) \quad (6)$$

3) *F Measure / F1 Score*

The value of the F1 Score is the weighted harmonic mean of precision and recall so that the highest value is one and the smallest is zero. . F1 Score emphasizes a balanced value between precision and recall. The formula is as follows:

$$F1Score = 2 * (Recall * Precision)/(Recall + Precision) \quad (7)$$

4) *Support*

This value is the actual number of class occurrences in the dataset. An unbalanced number between classes will worsen the prediction results of the classification [25].

III. Result and Discussion

This research leverages several libraries to support the data analysis process:

1) Pandas: Used for loading and preprocessing the pre-trained data.

2) Scikit-learn: Used to split the dataset into training and testing data.

3) Keras: Used for image augmentation with ImageDataGenerator and to import the pre-trained model.

4) TensorFlow: Used for fine-tuning the pre-trained model before training.

5) NumPy: Used for preprocessing the prediction results.

6) Matplotlib: Used to create visualizations like the confusion matrix, model accuracy plot, and model loss plot.

7) Seaborn: Used to create a heatmap on top of the visualizations generated by Matplotlib.

The results of the pre-trained models consist of the total time needed to train the model and prediction results based on the confusion matrix, accuracy, and f1 score from each model.

The test scenarios carried out are as follows:

1) ResNet50V2 pre-trained model with a batch size of 64
2) Xception pre-trained model with a batch size of 64
3) EfficientNetB0 pre-trained model with a batch size of 64
4) ResNet50V2 pre-trained model with a batch size of 32
5) Xception pre-trained model with a batch size of 32
6) EfficientNetB0 pre-trained model with a batch size of 32

A. Comparison of Results between Models with Batch Size 64

After doing scenario 1 until 3, results between models with a batch size of 64 by seeing from the perspective of the correctness of the prediction results can be seen in Table II. EfficientNetB0 is more accurate than other models.

TABLE II. Prediction Result of Each Model on the Test Dataset based on Confusion Matrix with a Batch Size of 64

	ResNet50 V2	**Xception**	**Efficient NetB0**
True Normal	195	185	196
False COVID-19	5	15	4
False Normal	28	26	13
True COVID-19	172	174	187

B. Comparison of Results between Models with Batch Size 32

After doing the scenario 4 to 6, results between models with a batch size of 32 by seeing from the perspective of the correctness of the prediction results can be seen in Table III. Using batch size 32, EfficientNetB0 also outperformed other models.

TABLE III. PREDICTION RESULT OF EACH MODEL ON THE TEST DATASET BASED ON CONFUSION MATRIX WITH A BATCH SIZE OF 32

	ResNet50 V2	Xception	Efficient NetB0
True Normal	190	187	193
False COVID-19	10	13	7
False Normal	51	79	35
True COVID-19	149	121	165

C. *Evaluation of Comparison Models with Batch Size 64 and 32*

The results of the comparison of the model with batch size of 64 and 32 can be seen in Table IV.

TABLE IV. COMPARISON OF MODEL RESULTS WITH BATCH SIZE 64 AND 32

Model	Batch Size	Accuracy	Training Time	F1 Score
2*ResNet50 V2	64	0.9175	3628s	0.9125
	32	0.8475	2252s	0.83
2*Xception	64	0.8975	7704s	0.8946
	32	0.77	3694s	0.7245
2*EfficientNetB0	64	0.9575	2553s	0.9565
	32	0.895	1923s	0.8871

From all of the evaluation metrics, the pre-trained model EfficientNetB0 with batch size 64 has the best accuracy of 0.9575 and F1 Score of 0.9565. The model with the fastest training time is EfficientNetB0 with a batch size of 32 for 1923 seconds. These results indicate that EfficientNetB0, which scales the width, depth, and resolution of convolutional neural networks with a fixed coefficient, is the best choice for COVID-19 infection diagnosis among ResNet50V2 and Xception.

IV. CONCLUSION

According to the research findings, the fine-tuning model with the best accuracy with a batch size of 64 is EfficientNetB0, which has an accuracy value of 0.9575, a training time of 2553 seconds, and an F1 score of 0.9565. Likewise, the best accuracy for the pre-trained model with a batch size of 32 is EfficientNetB0, which has an accuracy value of 0.895, a training time of 1923 seconds, and an F1 score of 0.8871.

As a result, it can be concluded that the EfficientNetB0 pretrained model with a batch size of 64 is the best application of the CNN algorithm among ResNet50V2, Xception, and EfficientNetB0 for the classification of COVID-19 infections in the COVIDx-CXR 2 dataset. The top outcomes are chosen because accuracy and F1 Score are more critical than training time. In the real-world scenario, training time only affects when the model is trained where accuracy and F1 Score affect image diagnosis at the time model will be used.

ACKNOWLEDGMENT

This research was fully funded by Universitas Multimedia Nusantara as a part of undergraduate thesis work.

V. CONCLUSIONS

In the results of research calculations that have been completed related to Indihome customer sentiment on Indihome services using the Naïve Bayes classification algorithm and Support Vector Machine to get the accuracy value, namely the accuracy of the Support Vector Machine algorithm is greater than the Naïve Bayes classification method. For this reason, in this study using 1000 Indihome customer datasets on the Twitter social media platform, the Support Vector Machine method is a better method than the Naïve Bayes method. Data is collected for 3 months starting from February 2021 to April 2021

However, this research still has several shortcomings, namely the process of labeling positive and negative sentiments is done manually which produces more negative sentiments than positive sentiments. There are differences from the data labeling that is applied manually to test the model using the class prediction results from the model classification results. In addition, this study only uses 1000 datasets. The accuracy of the Naive Bayes method is 82% while the Support Vector Machine is 84%

REFERENCES

[1] A. Japutra and R. Situmorang, “The repercussions and challenges of COVID-19 in the hotel industry: Potential strategies from a case study of Indonesia,” International Journal of Hospitality Management, vol. 95, p. 102890, May 2021.

[2] D. Tjahjana, D. Dwidienawati, A. H. Manurung, and D. Gandasari, “Does people’s wellbeing get impacted by COVID-19 pandemic measure in Indonesia?” Studies of Applied Economics, vol. 39, no. 4, May 2021. [Online]. Available: https://ojs.ual.es/ojs/index.php/eea/article/view/4873

[3] H. Putranto, “Covid-19 and the Crisis of Creative Industries in Digital Capitalism: Commodification of Digital Media Workers in the Framework of Data as Labor” Jurnal Politik, vol. 25, no. 2, pp. 9-48, April 2021. [Online]. Available: https://ejournal.atmajaya.ac.id/index.php/respons/article/view/2461

[4] E. Hidayat and D. Susilo, “Refusing to Die: Programmatic Goods in the Fight against COVID-19 in Sampang Regency” Respons: Jurnal Etika Sosial, vol. 7, no.1, pp. 47-73, March 2021. [Online]. Available: https://doi.org/10.7454/jp.v7i1.1001

[5] M. Rahimzadeh and A. Attar, “A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2,” Informatics in Medicine Unlocked, vol. 19, p. 100360, Jan. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914820302537

[6] WHO, “Coronavirus.” [Online]. Available: https://www.who.int/ westernpacific/health-topics/coronavirus

[7] I. Mporas and P. Naronglerdrit, “COVID-19 Identification from Chest X-Rays,” in 2020 International Conference on Biomedical Innovations and Applications (BIA). Varna, Bulgaria: IEEE, Sep. 2020, pp. 69–72. [Online]. Available: https://ieeexplore.ieee.org/document/9244509/

[8] Y. Laurensia, J.C. Young, A. Suryadibrata, “Early Detection of Diabetic Retinopathy Cases using Pre-trained EfficientNet and XGBoost,” International Journal of Advances in Soft Computing and its Applications, vol. 12, no. 3, pp 101-111, November 2020. [Online]. Available: https://www.i-csrs.org/Volumes/ijasca/2020.3.8.pdf

[9] K. A. Prayogo, A. Suryadibrata, J. C. Young, “Classification of pneumonia from X-ray images using siamese convolutional network,” TELKOMNIKA Telecommunication, Computing, Electronics and Control, vol. 18, No. 3, pp 1302-1309, June 2020. [Online]. Available : http://telkomnika.uad.ac.id/index.php/TELKOMNIKA/article/viewFile/14751/8474#:~:text=In%20this%20research%2C%20we%20used,bacterial%20pneumonia%2C%20and%20viral%20pneumonia.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” arXiv:1512.03385 [cs], Dec. 2015, arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385

[11] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” arXiv:1610.02357 [cs], Apr. 2017, arXiv: 1610.02357. [Online]. Available: http://arxiv.org/abs/1610.02357

[12] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” arXiv:1905.11946 [cs, stat], Sep. 2020, arXiv: 1905.11946. [Online]. Available: http://arxiv.org/abs/1905.11946

[13] E. Martinez Chamorro, A. Diez Tascon, L. Ibanez Sanz, S. Ossaba Velez, and S. Borruel Nacenta, “Radiologic diagnosis of patients with COVID-19,” Radiologia (English Edition), vol. 63, no. 1, pp. 56–73, Jan. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2173510721000033

[14] H. Gunraj, A. Sabri, D. Koff, and A. Wong, “COVID-Net CT-2: Enhanced Deep Neural Networks for Detection of COVID-19 from Chest CT Images Through Bigger, More Diverse Learning,” arXiv:2101.07433 [cs, eess], Jan. 2021, arXiv: 2101.07433. [Online]. Available : http://arxiv.org/abs/2101.07433

[15] S. T. Krishna and H. Kalluri, “Deep learning and transfer learning approaches for image classification,” 06 2019. [Online]. Available : https://www.semanticscholar.org/paper/Deep-learning-and-transfer-learning-approaches-for-Krishna-Kalluri/6c2a0e3a798d59655732980d2b03b9e89f586548ss

[16] Y. Liu and S. Ji, “A Multi-Stage Attentive Transfer Learning Framework for Improving COVID-19 Diagnosis,” arXiv:2101.05410 [cs, eess], Jan. 2021, arXiv: 2101.05410. [Online]. Available: http://arxiv.org/abs/2101.05410

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks,” arXiv:1603.05027 [cs], Jul. 2016, arXiv: 1603.05027. [Online]. Available: http://arxiv.org/abs/1603.05027

[18] M. Vakili, M. Ghamsari, and M. Rezaei, “Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification,” arXiv:2001.09636 [cs, stat], Jan. 2020, arXiv: 2001.09636. [Online]. Available: http://arxiv.org/abs/2001.09636

[19] J. D. Novakovic, A. Veljovic, S. S. Ilic, A. Papic, and T. Milica, “Evaluation of Classification Models in Machine Learning,” Theory and Applications of Mathematics & Computer Science, vol. 7, no. 1, pp. 39–46, Apr. 2017. [Online]. Available: https://www.uav.ro/applications/se/journal/index.php/TAMCS/article/view/158

[20] S. A. Hicks, I. Strumke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen, and S. Parasa, “On evaluation metrics for medical applications of artificial intelligence,” medRxiv, Tech. Rep., Apr. 2021, type: article. [Online]. Available: https://www.medrxiv.org/content/10.1101/2021.04.07.21254975v1

[21] A. C. Muller and S. Guido, Introduction to machine learning with Python: a guide for data scientists, first edition, Sebastopol, CA, 2016, oCLC: ocn895728667.

[22] R. Caruana, S. Lawrence, and L. Giles, “Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping,” in Proceedings of the 13th International Conference on Neural Information Processing Systems, ser. NIPS’00. Cambridge, MA, USA: MIT Press, Jan. 2000, pp. 381–387.

[23] D. Wilson and T. Martinez, “The need for small learning rates on large problems,” in IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222), vol. 1. Washington, DC, USA: IEEE, 2001, pp. 115–119. [Online]. Available: http://ieeexplore.ieee.org/document/939002/

[24] T. Fawcett, “An introduction to ROC analysis,” Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, Jun. 2006. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S016786550500303X

[25] D. M. W. Powers, “Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation,” Journal of Machine Learning Technologies, pp. 37–63, 2011. [Online]. Available: https://www.researchgate.net/publication/228529307