KNN Optimization Using Grid Search Algorithm for Preeclampsia Imbalance Class

Sukamto[1, 2, a),]Hadiyanto[1,3, b)] and Kurnianingsih[1, 2, c)]

Author Affiliations

[1]Doctoral Program of Information System, School of Postgraduate Studies Diponegoro University,50241 Semarang, Central Java, Indonesia
[2]Department of Electrical Engineering, Politeknik Negeri Semarang, 50275 Semarang, Central Java, Indonesia
[3]Department of Physics, Faculty of Science and Mathematics, Diponegoro University, 50241
Semarang, Central Java, Indonesia

Author Emails

a) Corresponding author: suk4mtho@students.undip.ac.id
b) Hadiyanto@live.undip.ac.id
c) kurnianingsih@polines.ac.id

Abstract

The performance of predicted models is greatly affected when the dataset is highly imbalanced and the sample size increases. Imbalanced training data have a major negative impact on performance. Currently, machine learning algorithms continue to be developed so that they can be optimized using various methods to produce the model with the best performance. One way of optimization with apply hyperparameter tuning. In classification, most of the algorithms have hyperparameters. One of the popular hyperparameter methodologies is Grid Search. GridSearch using Cross Validation makes it easy to test each model parameter without having to do manual validation one by one. In this study, we will use a method in hyperparameter optimization, namely Grid Search. The purpose of this study is to find out the best optimization of hyperparameters for two machine learning classification algorithms that are widely used to handle imbalanced data cases. Validation of the experimental results uses the mean cross-validation measurement metric. The experimental results show that the KNN model gets the best value compared to the Decision Tree.

Keywords: Machine learning, hyperparameters, grid search

I. INTRODUCTION

Preeclampsia is a leading cause of maternal death [1], the second leading cause of maternal death in the UK [2], in Europe [3], in several developing countries[3], and the main contributor to maternal mortality in the world[4], and also responsible for 75,000 maternal deaths worldwide each year[5]. Globally, 10–15% of all maternal deaths are attributable to preeclampsia or preeclampsia [6]. Maternal mortality is higher in developing countries than in developed countries [7], and up to 99% of these deaths occur in less developed countries [2].

The prediction of preeclampsia and its disorders has received much attention in the past two decades. Early detection and management of hypertension in pregnancy is required. The initial diagnosis of preeclampsia is based solely on blood pressure and urine tests for proteinuria and can be done remotely. Early diagnosis and treatment of high blood pressure can reduce maternal morbidity and consequently mortality.

In this international issue, medical personnel and researchers are looking for early detection of preeclampsia. The fast prediction and detection of preeclampsia disease is important not only for healthcare professionals to decrease the sum of preeclampsia disease patients but also for their patients. Several models for predicting the risk of preeclampsia have been done and validated in several studies. Class imbalance problems are predominant in the medical domain[10]. Medical data is often very unbalanced, namely the condition of the minority sample is far less than the majority sample [11].

In general, imbalanced datasets are a problem that is often found in health applications [12] In the classification of medical data, we often encounter a disproportionate number of data samples where at least one of the categories represents only a very small fraction of the data. At the same time, it is a difficult problem in most machine learning algorithms. There are many works dealing with the classification of unbalanced datasets[13]. Applying hyperparameter optimization for both classification algorithms and resampling approaches can produce the best results for classifying imbalanced datasets [14].

II. LITERATURE REVIEW

A. Decision Tree

Decision tree learning algorithms have been successfully used in expert systems to capture knowledge [15]. A decision tree is a type of data structure with nodes (also known as roots, branches, and leaves) and edges. To generate a decision tree, the Tree C4.5 method takes 4 steps in total. Initially, choose the characteristic as the root. Additionally, make a branch at each value. Set the dataset in the branch and, then. Until every layer has the same value, repeat step two a total of four times [16]. Decision Trees are the most appropriate approach to imbalance. Since the number of trees in each class is obviously highly imbalanced, the solution proposed in this paper is based on specific weights which correct the accuracy of the classification algorithms [17]. Decision trees, especially C4.5, are one of the most popular algorithms that have been greatly supported by sampling methods to combat high imbalances in class distributions [18].

B. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple and effective machine learning algorithm used for both classification and regression tasks. It is a non-parametric algorithm, meaning it doesn't make assumptions about the underlying data distribution. Instead, it makes predictions based on the similarity (distance) between the input data and the training examples.

The KNN algorithm is a commonly used method in classifiers based on individual learning. To solve the boundary overlapping problem, we design an edge sampling method based on K nearest neighbors [19] The KNN algorithm offers the foundation for other balancing techniques to address this issue. Studies have demonstrated that the KNN partial sampling method outperforms other sampling techniques. The proposed approach also outperforms the outcomes of other experiments, demonstrating how KNN's simplicity can serve as a foundation for effective machine learning and information-serving algorithms. [20]. Due to the simplicity of KNN (K-Nearest Neighbor) classification algorithm, it has been widely used in many field[21]. Although KNN is computationally expensive, it is very simple to understand, accurate, requires only a few parameters to be tuned and is robust [22].

C. Hyperparameter

hyper-parameters are adjusted for each model in order to find a hyper-parameter setting that maximizes the model performances and so that the ML model can predict unknown data accurately [23]. Hyperparameter plays an essential role in the fitting of supervised machine learning algorithms [24]. Every machine learning algorithm has a hyperparameter setting. Optimal hyperparameter helps in building a better machine learning model [25].The hyperparameter optimization problem can be represented by [26]. In order to effectively handle imbalanced data, hyperparameter tweaking with class weight optimization has been shown to be effective. Compared the performance of hyperparameter optimization with that of default hyperparameters for resampling and classification algorithms, and discovered that hyperparameter optimization can be more effective at classifying datasets, incomplete data. Provided a technique for hyperparameter adjustment to enhance model effectiveness for unbalanced data [23]. The computational time was dramatically reduced by up to 98.2% for the high-performance SVM hyperparameter tuning model, and the performance of cross-validation was also improved [27].

Hyperparameter tuning has a very important role in optimizing the performance of any machine learning algorithm. The value of the hyperparameter cannot be determined from the data and we always take it as given when defining the model, in other words, the hyperparameter value must be determined before a model undergoes its learning process. Hyperparameters are variables that affect the output of a model. In the k nearest neighbors model we know the hyperparameter k = (number of nearest neighbors), KNN with values k = 1 and k = 5 are likely to give different outputs even though they are given the same input. This study uses the Grid Search technique for the process of finding optimal hyperparameter values in the model. Grid Search Cross Validation is a term used to refer to Grid Search and Cross-Validation techniques, namely methods for selecting combinations of models and hyperparameters by testing each combination one by one and validating each combination [16]. Figure 2 is an illustrative example of the hyperparameter tuning process in the range 1-10 using the Grid Search Cross Validation.

The Next step, I would like show you about how Grid search Algorithm is work. And this part gives illustrated how Grid search is work. The grid search technique will construct many versions of the model with all possible combinations of hyperparameter and will return the best one. For a combination of K-fold and range of parameter, the performance score comes out to be highest, therefore it is selected.

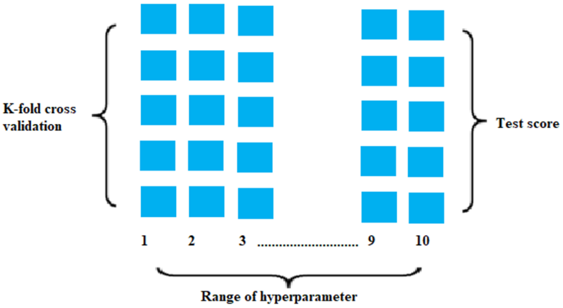

Figure 1. Hiperparameter metrics illustrative

One of the most crucial steps in a machine learning process is hyperparameter tweaking. The improper values for the hyperparameters could produce inaccurate results and a poorly performing model. Hyperparameter tweaking can be done in a variety of ways. Grid searches are one of them. Hyperparameters are model parameters whose values are set before training. A model with different hyperparameters is, actually, a different model so it may have a lower performance. If the model has several hyperparameters, we need to find the best combination of values of the hyperparameters searching in a multi-dimensional space. That's why hyperparameter tuning, which is the process of finding the right values of the hyperparameters, is a very complex and time-expensive task. Grid search is a combination of all the possible hyperparameter values used. Which creates a large matrix of hyperparameter combination set. This is reproducible and can be automated, but is not an efficient method for searching hyper-parameter space with a higher dimension. Grid search has been used extensively due to it's easy to implementation. It automatically go through the search space. In general grid search wastes resources exploring those parameter values which could be less important for a particular problem space [25].

III. RESULT AND DISCUSSION

The steps of the experiments applied to generate the results of this study were data preprocessing, exploratory data analysis (EDA), data preparation, model fitting (testing several models for hyperparameters), and result evaluation. The optimal hyperparameter among all the models was found.

A. Datasets

The dataset consists of several variables and one target variable (Diagnosis). Predictor variables include Age, Maternal arm circumference, Hb, and others. This is an example of an overview of preeclampsia statistics that may be seen in Table 1.

Table 1. Feature of Data Description

No	Feature Name	Description
1	Age	The mother's age at the time of delivery
2	Maternal arm circumference (MAC)	The number of months between the birth of the index child and the next live delivery
3	Hb	Hemoglobin
4	Systolic	Indicates the amount of pressure being exerted on the walls of your arteries when your heart beats
5	Diastolic	Indicates the amount of pressure being exerted on the walls of your arteries in between heartbeats
6	Protein in urine	A high level of proteins in the urine
7	Parity	Number of fetuses with a gestational age of 24 weeks or more after giving birth
8	Birth intervals	Number of months between first birth and next birth
9	Height	Height of body
10	Weight	Weight of body
11	History of PE (HPE)	Patient's mother has preeclampsia
12	History of DM (HDM)	The mother of the patient suffered from diabetes
13	History of hypertension (HH)	The mother of the patient suffered from hypertension
14	MAP	Mean Arterial Pressure
15	BMI	Body Mass Index
16	Diagnosis	This is the target attribute to classify

B. Data Preprocessing

The first step is to import some of the libraries needed in data pre-processing. This experiment uses the python3 language, so the libraries used include NumPy, pandas, matplotlib and seaborn. The next step is to load the dataset in CSV format using the panda's library. After the dataset is loaded, we have to check it and look for noise by creating a feature matrix X and a Y observation vector for X. The last step is to find and overcome missing values because they can cause problems during training. In figure 2, can be seen that dataset didn't have missing value.

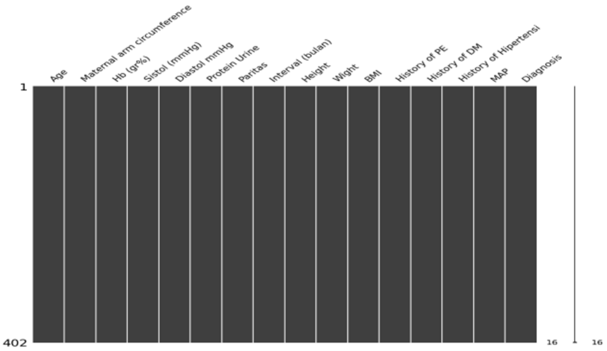

Figure 2. Missing value

C. Exploratory Data Analysis (EDA)

Exploratory data analysis at this stage was carried out to find the ratio of the imbalance class and see the relationship between the two data attributes, features that are closely related to preeclampsia.

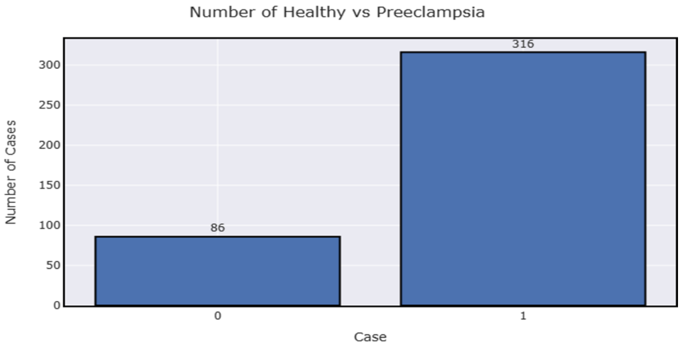

Figure 3. Imbalance Dataset

Figure 3 represent imbalance dataset, which X horizontal Axis represent kind of case, and Y vertical axis represent number of case. This graph show you that data in imbalance condition. Which is sum of preeclampsia only 86 people, meanwhile healthy people is 316 people. So, we need to do be balance condition, in this research we used Random under sampling. Since, it is an imbalanced dataset, I would first under sample the majority class (i.e. Preeclampsia in this case) so that I can get an equal distribution of Healthy and Preeclampsia cases. For undersampling I have used Random under sampling before crossvalidation.

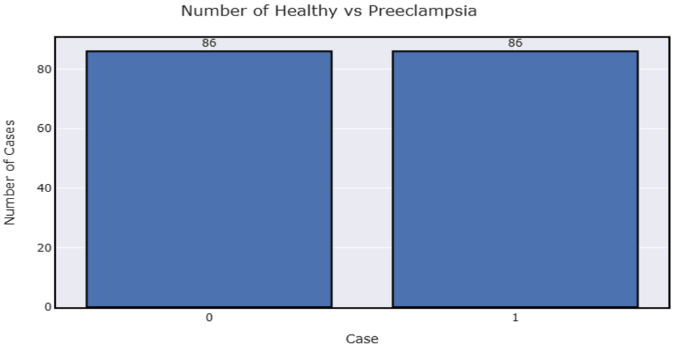

Figure 4. Balance Dataset

If you look at this figure 4, you will notice the data in balancing condition now, in which Healthy and Preeclampsia cases is equal namely 86. Figure 5 illustrates the correlation relationship between variables as illustrated by the seaborn correlation heatmap using python.

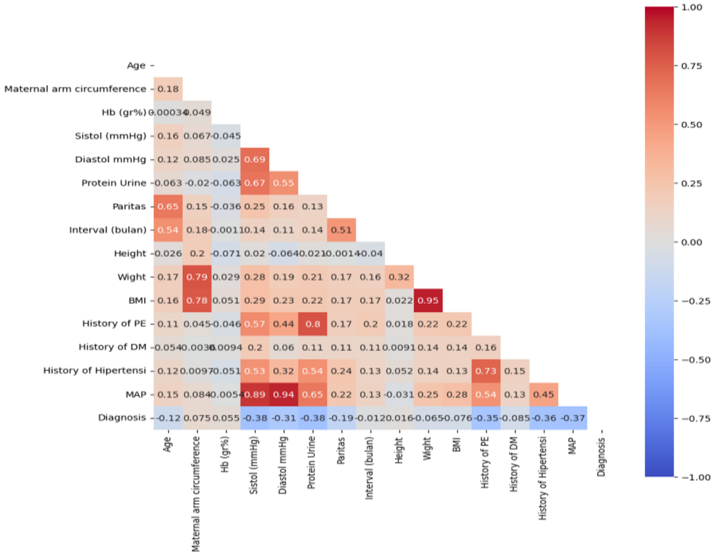

Figure 5. Heat Map Correlation

A correlation matrix is a tabular representation that displays the correlation coefficients among a group of variables. Its purpose is to identify the strength of connections between variable pairs. This matrix helps uncover both apparent and less apparent relationships among variables, proving valuable for researchers and analysts seeking insights into intricate variable relationships. Based on figure 5 it can be seen the correlation of each variable to the target.

Table 2. Benchmark Model

Type of Dataset	Accuracy Score	Precision	Recall	F1-score
Original	0.2139	0.2139	1.0000	0.3525
Undersampled	0.5000	0.5000	1.0000	0.6667

Table 3. Cross-Validation Scores

Model	Before Applying GridSearch	After Applying GridSearch
Decision Tree	60.53	62.05
KNN	61.96	63.45

Table 2 shows the test using the original sample and undersampled, the test shows that the undersample has better performance. Meanwhile, in table 3 the results of testing with Grid search, the KNN algorithm has a higher Cross validation score compared to the Decision Tree.

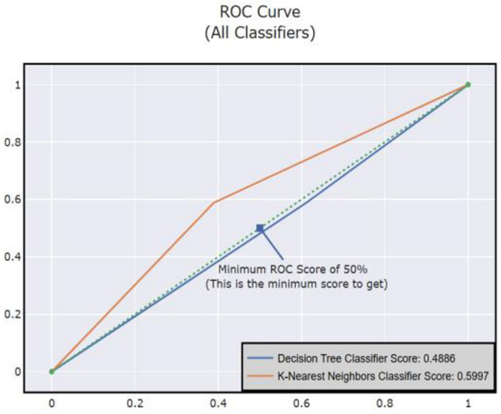

Figure 6. ROC curve

Figure 6 shows the ROC AUV analysis. The green line is the lower limit, and the area under that line is 0.5, and the perfect ROC Curve would have an area of 1. As closer our model's ROC AUC is to 1, the better it is in separating classes and making better predictions. By analogy, the higher the AUC, the better the model is at distinguishing between patients with healthy and preeclampsia.

IV. CONCLUSION

Based on research, it can be inferred that optimizing hyperparameters with Grid Search while using a machine learning model makes the process of building a model easier. Cross Validation, which is used by Grid Search, provides ease of use when evaluating each parameter of a model without the need for manual validation for each individual parameter. As we already know, grid search has a tuning process that requires time when hyperparameters are changed. This is because the number of possible parameter combinations increases rapidly with time, so it is essential to continue with this analysis.

ACKNOWLEDGMENT

This research was supported by the Doctoral Program of Information Systems, School of Postgraduate Studies, Diponegoro University, for supporting this research under Postgraduate Incentive Research Grant, Vote No.345-39/UN7.6.7/PP/2022 and Department of Electrical Engineering Politeknik Negeri Semarang Semarang.

REFERENCES

[1] L. C. Poon and K. H. Nicolaides, “Early Prediction of Preeclampsia,” *Obstet. Gynecol. Int.*, vol. 2014, no. Table 2, pp. 1–11, 2014, doi: 10.1155/2014/297397.

[2] P. Von Dadelszen and L. A. Magee, “Pre-eclampsia: An Update,” *Curr. Hypertens. Rep.*, vol. 16, no. 8, 2014, doi: 10.1007/s11906-014-0454-8.

[3] H. Sufriyana, Y. W. Wu, and E. C. Y. Su, “Artificial intelligence-assisted prediction of preeclampsia: Development and external validation of a nationwide health insurance dataset of the BPJS Kesehatan in Indonesia,” *EBioMedicine*, vol. 54, 2020, doi: 10.1016/j.ebiom.2020.102710.

[4] J. Zhang *et al.*, “Early prediction of preeclampsia and small-for-gestational-age via multi-marker model in Chinese pregnancies: A prospective screening study,” *BMC Pregnancy Childbirth*, vol. 19, no. 1, pp. 1–10, 2019, doi: 10.1186/s12884-019-2455-8.

[5] L. Myatt, “Expert Review The prediction of preeclampsia : the way forward,” *Am. J. Obstet. Gynecol.*, 2020, doi: 10.1016/j.ajog.2020.10.047.

[6] A. C. De Kat, J. Hirst, M. Woodward, S. Kennedy, and S. A. Peters, “Prediction models for preeclampsia: A systematic review,” *Pregnancy Hypertens.*, vol. 16, no. March, pp. 48–66, 2019, doi: 10.1016/j.preghy.2019.03.005.

[7] E. Purwanti and I. S. Preswari, “Early Risk Detection of Pre-eclampsia for Pregnant women using Artificial Neural Network,” vol. 15, no. 2, pp. 71–80, 2019.

[8] J. H. Jhee *et al.*, “Prediction model development of late-onset preeclampsia using machine learning-based methods,” pp. 1–12, 2019.

[9] J. Allotey *et al.*, “Development and validation of prediction models for risk of adverse outcomes in women with early-onset pre-eclampsia: protocol of the prospective cohort PREP study,” *Diagnostic Progn. Res.*, vol. 1, no. 1, pp. 1–8, 2017, doi: 10.1186/s41512-016-0004-8.

[10] M. A. Ganaie and M. Tanveer, “KNN weighted reduced universum twin SVM for class imbalance learning,” *Knowledge-Based Syst.*, vol. 245, p. 108578, 2022, doi: 10.1016/j.knosys.2022.108578.

[11] Z. Xu, D. Shen, T. Nie, Y. Kou, N. Yin, and X. Han, “A cluster-based oversampling algorithm combining SMOTE and k-means for imbalanced medical data,” *Inf. Sci. (Ny).*, vol. 572, pp. 574–589, 2021, doi: 10.1016/j.ins.2021.02.056.

[12] M. M. Rahman and D. N. Davis, “Addressing the Class Imbalance Problem in Medical Datasets,” *Int. J. Mach. Learn. Comput.*, no. May 2014, pp. 224–228, 2013, doi: 10.7763/ijmlc.2013.v3.307.

[13] S. Belarouci and M. A. Chikh, “Medical imbalanced data classification,” *Adv. Sci. Technol. Eng. Syst.*, vol. 2, no. 3, pp. 116–124, 2017, doi: 10.25046/aj020316.

[14] J. Kong, W. Kowalczyk, D. A. Nguyen, T. Back, and S. Menzel, “Hyperparameter Optimisation for Improving Classification under Class Imbalance,” *2019 IEEE Symp. Ser. Comput. Intell. SSCI 2019*, pp. 3072–3078, 2019, doi: 10.1109/SSCI44817.2019.9002679.

[15] S. Singh and P. Gupta, “Comparative Study Id3, Cart and C4.5 Decision Tree Algorithm: a Survey,” *Int. J. Adv. Inf. Sci. Technol. ISSN*, vol. 27, no. 27, pp. 97–103, 2014.

[16] E. Budiman, Haviluddin, N. Dengan, A. H. Kridalaksana, M. Wati, and Purnawansyah, “Performance of Decision Tree C4.5 Algorithm in Student Academic Evaluation,” *Lect. Notes Electr. Eng.*, vol. 488, no. February, pp. 380–389, 2018, doi: 10.1007/978-981-10-8276-4_36.

[17] C. O. Truică and C. A. Leordeanu, “Classification of an imbalanced data set using decision tree algorithms,” *UPB Sci. Bull. Ser. C Electr. Eng. Comput. Sci.*, vol. 79, no. 4, pp. 69–84, 2017.

[18] D. A. Cieslak and N. V. Chawla, “Learning decision trees for unbalanced data,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5211 LNAI, no. PART 1, pp. 241–256, 2008, doi: 10.1007/978-3-540-87479-9_34.

[19] H. Ding, L. Chen, L. Dong, Z. Fu, and X. Cui, “Imbalanced data classification: A KNN

and generative adversarial networks-based hybrid approach for intrusion detection," *Futur. Gener. Comput. Syst.*, vol. 131, pp. 240–254, 2022, doi: 10.1016/j.future.2022.01.026.

[20] M. Beckmann, N. F. F. Ebecken, and B. S. L. Pires de Lima, "A KNN Undersampling Approach for Data Balancing," *J. Intell. Learn. Syst. Appl.*, vol. 07, no. 04, pp. 104–116, 2015, doi: 10.4236/jilsa.2015.74010.

[21] W. Xing and Y. Bei, "Medical Health Big Data Classification Based on KNN Classification Algorithm," *IEEE Access*, vol. 8, pp. 28808–28819, 2020, doi: 10.1109/ACCESS.2019.2955754.

[22] H. Dubey and V. Pudi, "Class based weighted K-Nearest neighbor over imbalance dataset," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7819 LNAI, no. PART 2, pp. 305–316, 2013, doi: 10.1007/978-3-642-37456-2_26.

[23] R. Guido, M. C. Groccia, and D. Conforti, "A hyper-parameter tuning approach for cost-sensitive support vector machine classifiers," *Soft Comput.*, vol. 27, no. 18, pp. 12863–12881, 2022, doi: 10.1007/s00500-022-06768-8.

[24] H. Jin, "Hyperparameter Importance for Machine Learning Algorithms," pp. 1–8, 2022, [Online]. Available: http://arxiv.org/abs/2201.05132.

[25] B. Panda, "A survey on application of Population Based Algorithm on Hyperparameter Selection," no. April, 2020, doi: 10.13140/RG.2.2.11820.21128.

[26] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Adv. Neural Inf. Process. Syst. 24 25th Annu. Conf. Neural Inf. Process. Syst. 2011, NIPS 2011*, pp. 1–9, 2011.

[27] F. Zhang, M. Petersen, L. Johnson, J. Hall, and S. E. O'bryant, "Hyperparameter Tuning with High Performance Computing Machine Learning for Imbalanced Alzheimer's Disease Data," *Appl. Sci.*, vol. 12, no. 13, 2022, doi: 10.3390/app12136670.

[28] P. Kampstra, "V28C01-1," vol. 28, no. November, pp. 1–9, 2008, [Online]. Available: papers3://publication/uuid/692988CE-7E10-498E-96EC-E7A0CE3620A3.