JOURNAL OF
EMERGING INVESTIGATORS

Article

Comparing model-centric and data-centric approaches to determine the efficiency of data-centric AI

Hoa La[1], Kinh La[1]
[1] VNU-HCM High School for the Gifted, Ho Chi Minh, Vietnam

SUMMARY

In current machine learning approaches, data is crucial, yet it is often overlooked and mishandled in artificial intelligence (AI). As a result, many hours are wasted fine-tuning a model based on faulty data. Hence, there exists a new trend in AI, which is data-centric AI. We hypothesized that data-centric AI would improve the performance of a machine learning model. To test this hypothesis, three models (two model-centric approaches and one data-centric approach) were used. The model-centric approaches included basic data cleaning techniques and focused on the model, while the data-centric approach featured advanced data preparation techniques and basic model-training. We found that the data-centric approach gave a higher accuracy than the model-centric approaches. The model-centric approaches achieved 91% and 90% accuracy, respectively, whereas the data-centric approach achieved 97% accuracy.

INTRODUCTION

Artificial intelligence (AI) is a thriving field in computer science. The goal of AI, within research is to improve accuracy with the least resources. For a machine learning model, accuracy will increase over time - the higher number of training epochs (the greater amount of training time), the higher accuracy it achieves (1). However, at a certain point, its accuracy will no longer increase, posing a problem for researchers: how to continue to improve the model when model training is not effective anymore (2). A machine learning model can be broken down into two parts: the data and model (or the neural network structure). There is an analogy where data is described as the food and the model is described as the body. As they say, “we are what we eat”, so the model (or the body) will be “healthy” if it has good data but will be “unhealthy” with bad data.

Thus, a solution for the problem of model accuracy is to focus on the data quality. A model will improve if the data fed into it is of high quality (2). Therefore, machine learning researchers have been focusing on data preprocessing (3). In other words, data-centric AI is a new trend in AI. Besides advanced preprocessing techniques, data-centric AI requires high quality data labeling, which is the process of assigning one or more labels to the data (4). This involves both effort from data providers and machine learning operations (MLOps).

Model-centric AI is a subset of AI research that focuses on optimization, cost-function, and hyper-parameters (e.g., batch size, learning rate, number of layers and epochs) to improve models (5). Data-centric AI is another subset of AI that focuses on data preprocessing to improve the quality of the data which will eventually be fed into models (6). Data-centric AI requires data consistency while model-centric AI may accept inconsistent data labels (7). While the model-centric approach optimizes the model to deal with noisy data, the data-centric approach invests in data quality tools to clean noisy data (7). While the model-centric approach improves the model iteratively, a data-centric approach iterates the data quality (7).

There are many advantages of data-centric AI compared to model-centric AI. First, advances in models are assumed to reach a benchmark while advances in data still prove efficiency (2). Second, data-centric approaches allow for more domain experts to contribute as data are easier to understand to them than mathematical equations (2). Considering these advantages, we hypothesized that data-centric AI would improve the performance of a machine learning model. To test this hypothesis, a total of three machine learning models (two Model-centric and one Data-centric) were used. The two model-centric approaches focused on the neural network structures and the number of training epochs (iterations) to improve their accuracy, while the data-centric approach focused on refining the data with advanced techniques (outlier detection, feature creation, dimensionality reduction, feature scaling) to improve the accuracy. Our research revealed that the model-centric approaches attained an accuracy of 91% and 90%, while the data-centric approach had an accuracy of 97%. Given data-centric AI’s superiority to model-centric AI, more emphasis will be put on data quality and data preprocessing of machine learning.

RESULTS

We used the Wisconsin Diagnostic Breast Cancer dataset to compare model-centric and data-centric approaches (8). The two model-centric approaches included basic data cleaning techniques and model training for 60 epochs, whereas the data-centric approach featured advanced data preparation techniques (e.g., outlier detection, feature creation, and data balancing) and model training for only 20 epochs. Ten test runs (repetitions) were conducted.

After the model trainings and 10 test runs, the model-centric approach 1 and approach 2 achieved an average accuracy of 90.8% and 89.6%, respectively, while the data-centric approach achieved 96.6% accuracy (**Table 1**). Our results support that the data-centric approach performs better than model-centric approach 1 and approach 2 respectively with *p*-values of 0.00002 and 0.0008 both smaller than 0.05.

To describe the performance of a classification model on a set of test data, confusion matrices were used. The

	Model-centric approach 1	Model-centric approach 2	Data-centric approach
Test run #1	0.87	0.83	0.97
Test run #2	0.93	0.87	0.98
Test run #3	0.9	0.94	0.97
Test run #4	0.9	0.95	0.95
Test run #5	0.87	0.94	0.97
Test run #6	0.9	0.86	0.96
Test run #7	0.95	0.94	0.96
Test run #8	0.95	0.9	0.96
Test run #9	0.94	0.95	0.97
Test run #10	0.87	0.78	0.97
Mean	0.908	0.896	0.966
Standard Deviation	0.000531111	0.001746667	3.55556E-05
Variance	0.001062222	0.003493333	7.11111E-05

Table 1: Accuracy of the three model-building approaches (model-centric approach 1, model-centric approach 2, and data-centric approach) over 10 test runs (repetitions of the learning process).

model-centric approach 1 predicted 46 true negatives, 9 false positives, 4 false negatives, and 84 true positives among 143 predictions (**Table 2**). The model-centric approach 2 predicted 43 true negatives, 12 false positives, 4 false negatives, and 84 true positives among 143 predictions (**Table 3**). The data-centric approach predicted 65 true negatives, 3 false positives, 2 false negatives, and 66 true positives among 136 predictions (**Table 4**). In this particular breast cancer prediction problem, both model-centric approaches predicted an average of four false negatives among 143 data points (3% rate). This means that 3% of the time, the models could not detect the patient's cancer, which in a clinical setting could result in the patient not receiving treatment (**Table 2-3**). Compared to this number, the data-centric approach predicted 2 false negatives among 136 data points (1.4% rate), which demonstrates a significantly better result delivered in clinical setting (**Table 4**).

The accuracy increases during the whole training process of 3 models were also considered (**Figure 1**). This was obtained using Tensorboard, which is a visualization and tooling kit for machine learning. For model-centric approach 1 and approach 2, in the first epoch, the models only achieved accuracies of under 60%. After a training process of 60 epochs, the models gained an accuracy of 91% and 90%. For the data-centric approach, in the first epoch, the model achieved over 80% accuracy. After a training process of only 20 epochs, the model reached an accuracy of 97%. As can be clearly seen, refining data makes models function significantly better, from the beginning of the training process.

	Predicted: NO	Predicted: YES
Actual: NO	46	9
Actual: YES	4	84

Table 2: Confusion matrix of model-centric approach 1.

	Predicted: NO	Predicted: YES
Actual: NO	43	12
Actual: YES	4	84

Table 3: Confusion matrix of model-centric approach 2.

	Predicted: NO	Predicted: YES
Actual: NO	65	3
Actual: YES	2	66

Table 4: Confusion matrix of data-centric approach.

DISCUSSION

Our results suggest that data-centric AI could improve machine learning remarkably in terms of both accuracy and computing efficiency (number of epochs). At the end of the process, while the two model-centric approaches yielded around 90% accuracy after 60 epochs, the data-centric approach only took 20 epochs to attain 97% accuracy. In the research, we chose to use two model-centric approaches to help assess whether the results were generalizable and not specific to a model. And a total of ten test runs were used to strengthen the statistical robustness of the research as results may differ each time.

Comparing 3 approaches on a dataset, the importance of refining data is shown as the data-centric approach featuring several advanced data preprocessing techniques performed best. As people move towards data-centric AI, various impacts can be made. First, the work of machine learning is

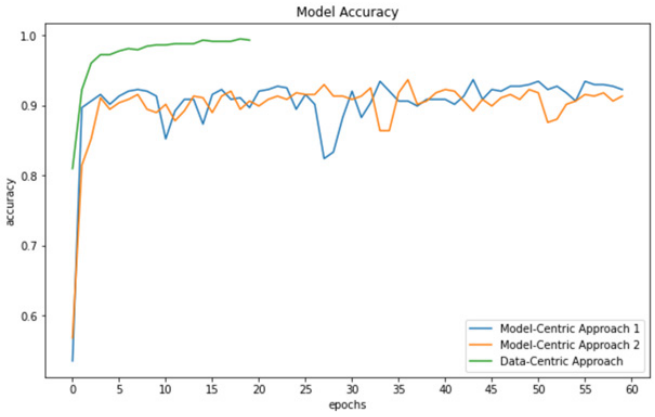

Figure 1: Accuracy of the three model approaches over time. After training the models, the model-centric approach 1, model-centric approach 2, and data-centric approach reached accuracies of 91%, 90%, and 97%, respectively.

80% preparing data and 20% developing the model (2). This is because time should be dedicated to preparing data as the most effective stage of the process rather than model training which delivered lower accuracy but consumed more computational and time resources. Secondly, as choosing and building a machine learning model no longer accounts for the principal part, machine learning models are commoditized (8). This means deep learning can be easily done through a single line of code: pip install. There exist several ML API services (e.g., AWS, Azure, GCE, Clarifai, and Bonsai) and AutoML tools (e.g., H20.ai, Auto-Keras, and Auto-sklearn) to help ease the process of applying machine learning. This helps spread the applications of machine learning and deep learning in various disciplines, ranging from health care to transportation, manufacturing to defense, and agriculture to retail (8). Thirdly, because data becomes the center of machine learning, various innovations about data collection, labeling, and management can be made, resulting in high-growth startups founded like Snorkel.ai and Scale.ai (9).

One limitation of our study is that there were no null values in the dataset. Hence, only a moderate difference in terms of accuracy was observed, not showcasing the ultimate strength of data-centric AI. Strategies to improve our research include using more tabular datasets and involving image and text datasets. More tabular datasets could be used to ensure the results can be generalizable given the limitation of our current dataset. And the usage of image and text datasets can help demonstrate data-centric AI efficiency on different data formats beyond numeric tabular ones. While data-centric AI works better than model-centric AI, data-centric AI does have some limitations. When the data becomes excessively cleaned after several stages of data preprocessing, some data points which are not noise may be wrongly classified as noise and thus removed. This can lead to overfitting and damage the model. Therefore, data preprocessing must be used within a certain limit to avoid model overfitting.

MATERIALS AND METHODS

Data

The Wisconsin Diagnostic Breast Cancer dataset contains measurements on cells in suspicious lumps removed from patient breasts (10). These measurements (e.g., radius, texture, perimeter, area, smoothness etc.), called features in the dataset, were computed from digitized images of fine needle aspirates (FNA) of breast masses. The measurements describe characteristics of the cell nuclei present in each image. All samples are classified as either benign(harmless) or malignant(harmful). This dataset is a table of 569x32 (row x column) in which the row represents the data points and the column represents the features (10).

Tools

A variety of tools were used for this research: Colab, Scikit-learn, Keras, and Tensorflow (11-13). Google Colaboratory (Colab), which is a product from Google Research, allows users to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs (14, 15). Scikit-learn (Sklearn) is a free machine learning library for Python (16). It features various classification, regression and clustering algorithms and eases machine learning uses. Keras is an open-source software library that provides a high-level Python interface for deep learning. Tensorboard is Tensorflow's visualization toolkit was used to access the performance of the models during training.

Model-centric AI approaches

Despite being non-data-centric AI approaches, basic steps of data preprocessing were required in order to feed the data accurately into the models. First, all unnecessary columns (e.g., ID, Unnamed) were dropped. Next, as machine learning models can only understand numeric inputs, diagnosis results were converted into numbers by using the function pandas.get_dummies (17). More specifically, "malignant" and "benign" were converted into 0 and 1, respectively. After preprocessing data, there were 30 features (columns) for 569 samples (rows). The data was then randomly split using an 80/20 training/testing paradigm, executed using Scikit-learn's train_test_split function.

Deep learning works based on an artificial neural network (ANN) were created. An ANN is comprised of 3 layers of neurons: the Input Layer, the Hidden Layer(s), and the Output Layer. As the importance of the input features varies, connections between neurons in 2 consecutive layers were associated with a weight. Then, an activation function was applied to the data to standardize the output of the neuron. Iterating through the dataset produced a cost function, which demonstrated the difference between the predicted outputs and the true outputs. To minimize this cost function, the weights between neurons were changed using gradient descent after every iteration (also called an epoch) (18).

All samples were categorized as either malignant or benign, which indicated a binary classification problem. Two deep learning models (artificial neural networks) would be used to represent model-centric AI. The neural network of the model-centric approach 1, built with Tensorflow and Keras, consisted of five layers. The input layer consisted of 30 nodes as there were 30 features after data preprocessing. The three hidden layers consisted of 50, 30, and 20 nodes. The output layer consisted of 1 node. The model was compiled with the optimizer Adam and the loss function Binary Crossentropy. It was then trained with 60 epochs (iterations) to provide the final accuracy. The neural network of the model-centric approach 2 was more complicated than that of the model-centric approach 1, consisting of 7 layers. The input layer consisted of 30 nodes. The hidden layers consisted of 60, 50, 40, 30, and 30 nodes. The output layer consisted of 1 node. The model was also compiled with the optimizer Adam and loss function Binary Crossentropy and trained with 60 epochs.

Data-centric AI approach

In the data-centric AI approach, emphasis was placed on data preprocessing rather than building and training models. Data preprocessing would contain more steps in addition to steps of the first approach: outlier detection, feature engineering, and balancing dataset.

An outlier is an observation that deviates drastically from other observations in the dataset. Causes include natural conditions (e.g. Elon Musk's income for income) and typing errors (e.g. human's weight of 1000 kg due to mistyping an extra 0 for weight). Outliers are detected and dropped by using Tukey IQR techniques. Defined by Tukey, outliers are values

more than 1.5 times the interquartile range from the quartiles (19).

Feature engineering is the process of making changes to the features of a dataset to improve machine learning model training. Feature engineering consists of 4 steps: feature creation, feature transformation (feature scaling), feature extraction, and feature selection (20).

Usually, in a dataset, there are relationships between features (e.g. height may be related to weight). In order to harness these relationships, we can use feature creation, which involves creating new features by using interactions between existing ones. A simple two-way interaction is represented by X3 = X1*X2 where X3 is the interaction between X1 and X2 where X1 and X2 are 2 different features in the dataset. Before applying feature creation, there were 30 features in the dataset. The number of interactions between these 30 features would be 435 (30*29/2). Then, after this step, there were a total of 465 (435+30) features.

Feature scaling normalizes the range of features of the data. By feature scaling, the gradient descent converges smoother, improving the model and reducing the training time (21). Feature scaling uses a technique called standardization, which centers the values around the mean with a unit standard deviation (21). Standardization does not change the number of features, which was 465.

Feature extraction reduces redundant data from the data set, which reduces training time tremendously. Principal component analysis (PCA) is used in this step. PCA is a technique that transforms a dataset of many features into principal components that summarize the variance that underlies the data. Each principal component is calculated by finding the linear combination of features that maximizes variance, while also ensuring zero correlation with the previously calculated principal components (22). After applying PCA, the number of features reduced from 465 to 10.

Feature selection chooses a subset of relevant features for use in a model construction. Feature selection is different from dimensionality reduction. Both methods function to decrease the number of attributes in the dataset; however, dimensionality reduction creates new groupings of attributes whereas feature selection includes and removes attributes without modifying them (23). Among those 10 features, 8 features were selected and remained in the dataset.

In this dataset, there were 282 benign and 173 malignant samples. As this imbalance could have resulted in model bias, balancing the dataset was essential. Synthetic Minority Oversampling Technique (SMOTE) was used. SMOTE randomly increases minority class examples by replicating them. SMOTE synthesizes new minority instances between existing minority instances and produces the virtual training records by linear interpolation for the minority class (24). After applying SMOTE, the number of benign samples equaled the number of malignant samples at 282 samples. The strategy for splitting the dataset for the data-centric approach was the same as that of model-centric approaches.

The neural network of the data-centric approach resembled that of the model-centric approach 1, consisting of 5 layers. The input layer consisted of 30 nodes. The hidden layers consisted of 50, 30, and 20 nodes. The output layer consisted of 1 node. The model was also compiled with the optimizer Adam and loss function Binary Crossentropy but only trained with 20 epochs.

ACKNOWLEDGEMENTS

We would like to thank UCI Machine Learning Repository for supplying the WDBC dataset. This research would not have been possible without the data for machine learning. We would also like to express our gratitude to our dad as a mentor for this research.

Received: March 13, 2022
Accepted: August 29, 2022
Published: April 20, 2023

REFERENCES

1. Gupta, Suyog, *et al*. "Model Accuracy and Runtime Tradeoff in Distributed Deep Learning:A Systematic Study." 2016. arxiv.org/abs/1509.04210.
2. Ng, Andrew. "A Chat with Andrew on MLOps: From Model-centric to Data-centric AI". Youtube, uploaded by DeepLearningAI, 24 Mar. 2021, www.youtube.com/watch?v=06-AZXmwHjo
3. Li, Canchen. "Preprocessing Methods and Pipelines of Data Mining: An Overview." 2019. arxiv.org/abs/1906.08510
4. Hendrycks, Dan, *et al*. "Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise." 2019. arxiv.org/abs/1802.05300
5. Hamid, Oussama H. "From Model-Centric to Data-Centric AI: A Paradigm Shift or Rather a Complementary Approach?" *2022 8th International Conference on Information Technology Trends (ITT)*, 2022, doi:10.1109/itt56123.2022.9863935
6. Jakubik, Johannes, *et al*. "Data-centric Artificial Intelligence." 2022. arxiv.org/abs/2212.11854
7. Zha, Daochen, *et al*. "Data-Centric AI: Perspectives and Challenges." 2023. arxiv.org/abs/2301.04819
8. Stoica, Ion, *et al*. "A Berkeley View of Systems Challenges for AI." 2017. arxiv.org/abs/1712.05855
9. Winecoff, Amy A., and Elizabeth Anne Watkins. "Artificial Concepts of Artificial Intelligence: Institutional Compliance and Resistance in AI Startups." 2022. arxiv.org/abs/2203.01157.
10. Wolberg, W. H., Street, W. N., and Mangasarian, O. L. Breast Cancer Wisconsin (Diagnostic) Data Set. Irvine, CA: UCI Machine Learning Repository, 1995. Web. 15 Aug 2022. www.kaggle.com/uciml/breast-cancer-wisconsin-data
11. Pedregosa, Fabian, *et al*. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research*, vol. 12, no. Oct, 2011, pp. 2825–2830, jmlr.org/papers/v12/pedregosa11a.html.
12. Chollet, Francois *et al*. Keras. keras.io, 2015.
13. Abadi, Martín, *et al*. "TensorFlow: A System for Large-Scale Machine Learning." 2016. arxiv.org/abs/1605.08695
14. Kluyver, Thomas *et al*. "Jupyter Notebooks - a publishing format for reproducible computational workflows." *ELPUB* (2016).
15. Fernando Pérez, Brian E. Granger, "IPython: A System for Interactive Scientific Computing", Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, May/June 2007, doi:10.1109/MCSE.2007.53.
16. Guido Van Rossum and Fred L Drake Jr. Python reference manual. Centrum voor Wiskunde

17. The pandas development team. Pandas-dev/pandas: Pandas, February 2020.
18. Roberts, Daniel A, *et al*. "The Principles of Deep Learning Theory: An Effective Theory Approach to Understanding Neural Networks." New York, Cambridge University Press, 2022.
19. Tukey, John W. "Exploratory Data Analysis." Reading, Mass., Addison-Wesley Pub. Co, 1977.
20. HEAVY.AI. "What Is Feature Engineering? Definition and FAQs | HEAVY.AI." www.heavy.ai, 2022, www.heavy.ai/technical-glossary/feature-engineering#:~:text=Feature%20engineering%20in%20ML%20consists. Accessed 15 Aug. 2022.
21. Wan, Xing. "Influence of Feature Scaling on Convergence of Gradient Iterative Algorithm." Journal of Physics: Conference Series, vol. 1213, no. 3, June 2019, p. 032021, 10.1088/1742-6596/1213/3/032021.
22. Abdi, Hervé, and Lynne J. Williams. "Principal Component Analysis." Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, no. 4, 30 June 2010, pp. 433–459, 10.1002/wics.101.
23. Guyon, Isabelle, and André Elisseeff. "An Introduction to Variable and Feature Selection." Journal of Machine Learning Research, vol. 3, no. Mar, 2003, pp. 1157–1182, www.jmlr.org/papers/v3/guyon03a.html.
24. Chawla, N. V., *et al*. "SMOTE: Synthetic Minority Over-Sampling Technique." Journal of Artificial Intelligence Research, vol. 16, no. 16, 1 June 2002, pp. 321–357, 10.1613/jair.953.