SAND2022-12576C

Measuring Reproduciblity of Machine Learning Methods for Medical Diagnosis

1st Hana Ahmed
Sandia National Laboratories
Albuquerque, USA
hahmed@sandia.gov

2nd Roselyne Tchoua
DePaul University
Chicago, USA
rtchoua@depaul.edu

3rd Jay Lofstead
Sandia National Laboratories
Albuquerque, USA
gflofst@sandia.gov

***Abstract*—The National Academy of Sciences, Engineering, and Medicine (NASEM) defines *reproducibility* as "obtaining consistent computational results using the same input data, computational steps, methods, code, and conditions of analysis," and *replicability* as "obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data" [1]. Due to an increasing number of applications of artificial intelligence and machine learning (AI/ML) to fields such as healthcare and digital medicine, there is a growing need for verifiable AI/ML results, and therefore reproducible research and replicable experiments. This paper establishes examples of irreproducible AI/ML applications to medical sciences and quantifies the variance of common AI/ML models (Artificial Neural Network, Naïve Bayes classifier, and Random Forest classifiers) for tasks on medical data sets.**

***Index Terms*—machine learning, reproducibility, randomness, pseudo-random number generator, neural network, naïve Bayes, random forest**

I. Introduction

With the increasing use of artificial intelligence (AI) and machine learning (ML) methods to analyze complex medical datasets, including medical disease diagnosis and treatment [2]–[4], it is essential to ensure that studies focused on the AI/ML application meet the same standards expected of medical research papers. That is, they publish their data and others can arrive at the same conclusions with their own analysis. In the case of ML models, equivalent models with the same accuracy presented should be able to be recreated using the same algorithms, data, and runtime settings. Some papers strive to achieve this by meticulously documenting the algorithm and settings to what the authors believe is a sufficiently detailed level. Others try hard, but inadvertently leave out critical information necessary to achieve the same results independently.

Publishing datasets has been a hit or miss effort that has led to the creation of the FAIR Data Principles [5]. In summary, FAIR stands for Findable, Accessible, Interoperable, and Reusable, defining four principles widely agreed upon for trustworthy science. A critical goal of FAIR data management and a benchmark for trustworthy science is experimental reproducibility. The majority of scientists will fail to reproduce findings of a prior study [6], informing what is considered a "reproducibility crisis" across scientific fields. Many scientific findings are in fact the results of repeated retrial and retesting until the desired results are achieved, without emphasizing that the results are actually extremely rare and unlikely to obtain by running the same experiment [7].

The wide availability of common data sets both from the University of California, Irvine Machine Learning data set archive as well as data published seeking to achieve FAIR standards for either a single publication or some larger research effort enables practitioners to re-attempt the published experiments relatively easily to better understand the work presented. Even with the more complex models generated using ML techniques compared to older analysis techniques, the reproducibility standard should still hold. However, the additional information necessary to properly reproduce results is not clearly understood and inconsistent practices have led to good work not being reproducible lending doubt to the accuracy.

With randomization present in essentially all ML model generation, the simplest standard of publishing the random number source and seed is often overlooked. ML model reproducibility can only be achieved by ensuring the reproducibility of psuedo-random number sequences generated during model training. Psuedo-random number sequences are produced by pseudo-random number generators (PRNGs) and can be replicated using identical PRNG seed values (the starting value of a pseudo-random number sequence). This paper contributes to the conversation of FAIR and reproducible scientific research by demonstrating that published ML models for medical decision making can have inconsistent performance due to a combination of uncontrolled randomness and insufficient model design information accompanying published results.

Even beyond the PRNGs, other factors, such as training batch sizes, must be recorded and published for someone else to be able to reproduce the work. Missing other parameters or using a subset of available data without clear justification can make a seemingly excellent study raise subtle doubts. These doubts may or may not be justified, but the missing additional information should prompt readers to ask, "why was this left out?"

This paper has selected two excellent published studies using ML for data analysis. The first uses n Artificial Neural Network (ANN) to achieve a high accuracy score in analyzing a single data set. The second uses three different algorithms on common data, but only a portion without justifying that

selection. We attempt to recreate the results presented in both of these studies and offer lessons learned for how to best document ML model generation and experiments to enable reproducibile results as well as achieve FAIR data principles.

The rest of the paper is organized as follows. First Section II is an overview of related work. Next, Section III is an overview of the data sets evaluated in the studies under evaluation in this work. Experiments attempting to reproduce the selected works are presented in Section IV followed by a discussion in Section V. Finally, in Section VI, conclusions are presented.

II. Related Work

Scholars in the medical and health sciences are increasingly calling for standardization of FAIR scientific research and data management [8]–[10]. Basareh et al. [11] demonstrates on the Hospital Averse Incidents Classification Scheme (HAICS) data set that FAIR ontologies contribute to the accountability of and transparency of ontology-based AI systems. Furthermore, Hooker et al. [12] and Yona et al. [13] argue for the importance of ensuring fairness in algorithm design in addition to fairness in data set collection, and address the potential implications of optimizing algorithm parameters for maximum accuracy score.

Carter et al. [14] addresses the need for reproducible AI research to ensure replicability across hardware systems, as computational variances can cause inconsistency even when using identical raw data, programming code, and computing environments. Zhuang et al. [15] quantifies the impact of system noise and the cost of ensuring determinism for various hardware types. Carter et al. also notes the irreproducibility of psuedo-random number sequences on high-performance computing systems. As a solution to this problem, Salmon et al. [16] introduces a series of high quality, parallelizable PRNGs—called AES, Threefish, and Philox—that can guarantee reproducible psuedo-random number sequences across CPUs, GPUs, clusters, and special-purpose hardware.

Randomness in neural networks and the importance has also been investigated by Scardapane et al. [17]. Pham et al. [18] and Alahmari et al. [19] study the repeatability of deep learning models using Python's TensorFlow and Keras libraries given the presence of uncontrolled randomness. Pham et al. studies the variance in accuracy scores of deep learning training algorithms given changes in training data, algorithm, network, and PRNG seed. Alahmari et al. finds that while deep learning models can be repeated by saving the random initializations, computational variances may results in a different learning process for a given run.

In the past, computer scientists have regenerated pseudo-random number sequences by recording the PRNG seed. Sen et al. [20] uses a "capture-and-replay" method to replay data races using the same PRNG seed. Frederickson et al. [21] uses a similar approach to reproduce Monte Carlo trees. Ahmed et al. [22] uses this approach in a series of experiments showing the extent to which the PRNG seed, train/test split ratio, and train and test data sets can impact final accuracy scores of neural networks, k-means clustering, and Naïve Bayes classifier models on scientific data sets. Ahmed et al. further shows that variance in accuracy score can be exacerbated or diminished depending on ML algorithm type and data set. Our paper extends on this line of work by investigating the possible variance of popular ML algorithms for medical diagnosis, and establishing examples of irreprodicible published scientific results.

III. Data Sets

This research utilizes a total of five benchmark data sets. Experiment 1 (Section IV-A) evaluates the predictive performance of ANNs on the *Pima Indian Diabetes* (PID) data set collected by the National Institute of Diabetes and Digestive and Kidney Diseases [23]. The PID data set consists of 768 instances and nine features: pregnancies, glucose, blood pressure, skin thickness, insulin, body mass index (BMI), diabetes pedigree function, age. The last feature is the target class label. We follow the preprocessing steps used by Khanam et al. [24] in their comparison of ML algorithms for diabetes prediction. Preprocessing was done using the WEKA data mining software tool, and consisted of missing value identification, outlier rejection, feature selection, and normalization. Using WEKA, we identified missing values, and identified and removed outliers and extreme values based on interquartile ranges. While Khanam et al. reports finding numerous missing values in the PID data set, we identified 0 missing values using WEKA. Additionally, Khanam et al. reports finding 45 outliers and 26 extreme values using WEKA, while we found 49 outliers and 0 extreme values. This reduced the PID data set to 719 instances. Then, based on the Pearson correlation coefficient feature selection method described by Khanam et al., we removed three features that were measurably irrelevant to the final outcome: skin thickness, blood pressure, and diabetes pedigree function. This left five predictive features: pregnancies, glucose, insulin, BMI, and age.

In Experiment 2 (Section IV-B), the *Heart Disease* database is used to compare the predictive performances of Gaussian Naïve Bayes classifiers, Bernoulli Naïve Bayes classifiers, and Random Forest classifiers. *Heart Disease* consists of four data sets collected from heart disease patients at four different medical centers: the *Cleveland Heart Disease* data set from the Cleveland Clinic Foundation; the *Switzerland Heart Disease* data set from the University Hospitals in Basel and Zurich; the *Hungary Heart Disease* data set from the Hungarian Institute of Cardiology in Budapest; and the *Long Beach Heart Disease* data set from the Veterans' Affairs (V.A.) Medical Center in Long Beach. We used the processed versions of these data sets made available in the UCI ML Repository [25] to train and test each ML model. Although Bernando et al [26]. did not detail how they dealt with missing values during preprocessing, we replaced missing values in each *Heart Disease* data set with the mean of the column values. Each *Heart Disease* data set varies in number of instances, but share the same 14 features: age, sex, chest pain type (cp), resting blood pressure (trestbps), serum cholestoral (chol), fasting blood sugar (fbs), resting electrocardiographic results (restecg), maximum heart rate achieved (thalach), exercise induced angina (exang), ST

depression induced by exercise relative to rest (oldpeak), slope of the peak exercise ST segment, number of major vessels colored by flourosopy (ca), thalassemia (thal). The last feature is the target class label.

IV. Experiments

This paper consists of two sets of experiments, each investigating the results of published ML algorithms on public medical data sets. In Experiment 1, we use a Philox PRNG made available in Python's Keras library. Philox is considered a very high quality generator due to its speed, range, and period, and it is guaranteed reproducibility for parallel and GPU systems [15]. Experiment 2 relies on Python's default Mersenne Twister generator, a less robust but still high quality PRNG [27].

Experiment 1 aims to reproduce the ANN model for diabetes prediction published by Khanam et al. [24]. Experiment 2 aims to reproduce the various ML models for heart disease prediction published by Bernando et al. [**?**].

A. Experiment 1: Predicting diabetes with Artificial Neural Networks (ANNs)

There have been numerous studies publishing the results of ANN models for diabetes prediction, with several ANN models reaching accuracy scores of over 80% on diabetes data sets [24], [28], [29]. Our first experiment quantifies the variance in diabetes prediction accuracy of ANNs with two hidden layers. We follow the data preprocessing and model construction methods of Khanam et al. [24], whose ANN model achieved an accuracy score of 88.6%. Our ANN algorithm was developed in Python using the Sequential class in Keras and TensorFlow, and using the Philox PRNG for Keras. The ANN consists of four dense layers, where the first and fourth are input and output layers, respectively. The input layer consists of five neurons (for five input features) and uses the RELU activation function. The second dense layer consists of five neurons, and the third consists of 26 neurons. Both hidden layers also use the RELU activation function. The output layer has only one neuron to output a binary value (0 to indicate no heart disease, and 1 to indicate heart disease), and uses the sigmoid activation function.

Using this algorithm, we generate 100 ANN models on the PID data set. For each model, a datetime method is used to set a unique PRNG seed. As per Khanam et al., models are fitted and tested with an 85:25 train/test split, and trained on 400 epochs with a learning rate of .01. We noted that the batch size used by Khanam et al. is not specified in their publication and there is no consensus within the ML community as to finding the optimal batch size for a given algorithm and data set. In fact, research has shown that higher batch sizes increase accuracy scores for convolutional neural networks [30], whereas lower batch sizes increase accuracy scores for deep neural networks [31]. Because we have no reliable method for determining the best batch size for our ANN, we chose an arbitrary batch size of 64 for the initial round of tests, for which results are shown in Fig. 1.

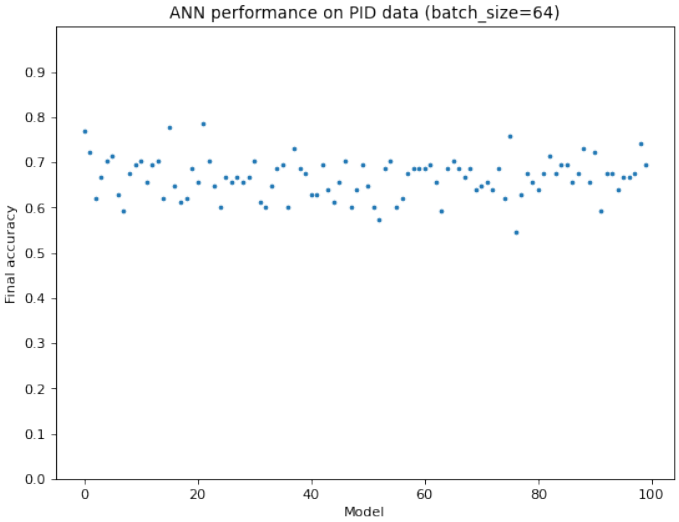


Fig. 1: 100 ANNs generated on the PID data set, where each model has a unique PRNG seed.

Across the 100 different ANN models, the best performing ANN had an accuracy score of 78.7%, and the worst performing ANN has an accuracy score of 54.63%, making for an overall difference of 24.07% accuracy points between the best and worst ANN models.

We did not achieve Khanam et al.'s accuracy score of 88.6%, and while a high accuracy may be attributed to a certain PRNG seed, we also considered that a different batch size could increase model accuracy. So, we carried out another set of tests, where we generated ANN models with a fixed PRNG seed (specifically, the seed value that was set for the ANN model with a 78.7% final accuracy score) and at every possible batch size (i.e., every integer in range 1 to 719, the size of the data set). This resulted in 719 models, each with a final accuracy score of 78.7%, which suggested that the training batch size wasn't significantly impactful on the final performance of an ANN model.

However, after correspondence with the authors, we learned that the batch size used by Khanam et al. was 1. Using a batch size of 1 in neural network training is called online or incremental training (as opposed to batch training, where the batch size is equal to the size of the entire training data set). Online training is typically used when a learning algorithm's knowledge base is being continuously updated with new training samples [32]. Online training is shown to result in faster convergence and lower computational cost with no significant difference in testing accuracy compared to training on larger batch sizes [33], [34]. We continued this experiment by generating 100 ANN models on the PID data set using a different PRNG seed and a batch size of 1 each. Results are shown in Fig 2.

When the PRNG seed is varied and the batch size is set to 1, we find that the best performing ANN model has an accuracy score of 100% whereas the worst has 62.96% accuracy, making for a difference of 37.04% accuracy points

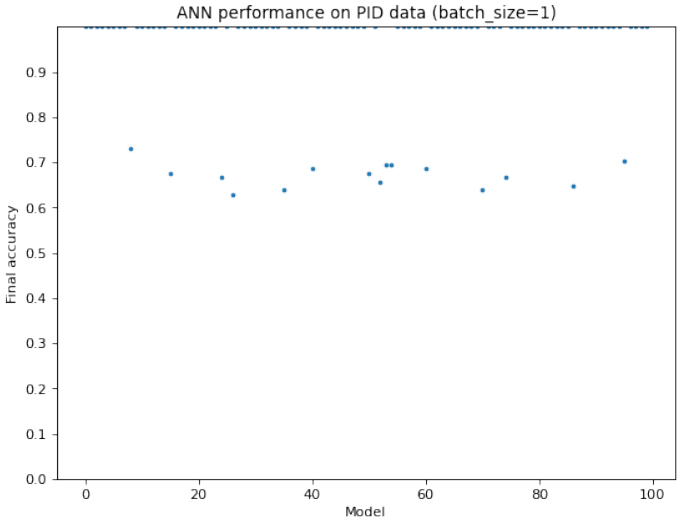


Fig. 2: 100 ANNs generated on the PID data set, where each model uses a different PRNG seed and a batch size of 1.

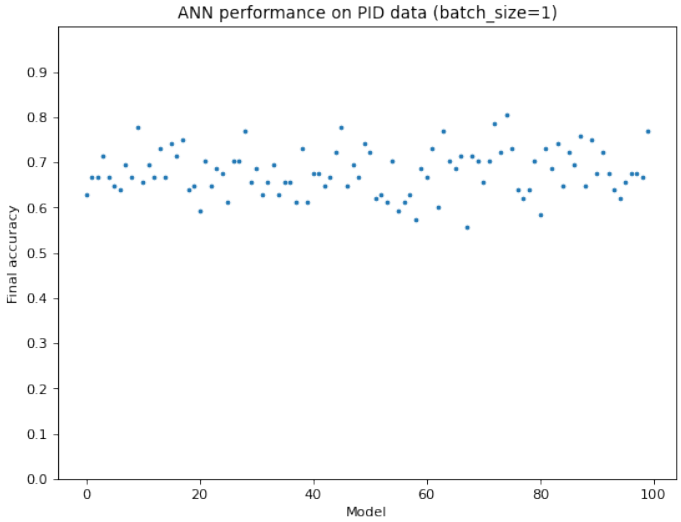


Fig. 3: 100 ANNs generated on the un-normalized PID data set, where each model uses a different PRNG seed and a batch size of 1.

between the best and worst ANN models. Although many of the ANN models score 100% accuracy, as seen in Fig 2, the significant drops in accuracy points further demonstrate the consequence of PRNG seed on final model outcome. We were still unable to achieve the result of 88.6% published by Khanam et al., demonstrating the difficulty of replicating a model guaranteeing exact performance. Whether this variance is due to differences in PRNG seed, data preprocessing, or unintentional deviation from Khanam et al's experiment design is unclear. But the results shown in Fig 1 and Fig 2 contradict the notion that batch size does not affect final model accuracy score, as reducing the batch size from 64 to 1 improved the accuracy score of our ANN models by as much as 45.37% points.

Furthermore, we noted that in Khanam et al., the reason given for normalizing the PID data set during preprocessing was to speed up computation. However, there is reason to suggest that normalizing a data set prior to training and testing actually increases ANN model accuracy [35]. So, in our next set of tests (Fig 3), we generate 100 ANN models using different seeds and a batch size of 1 on an un-normalized PID data set.

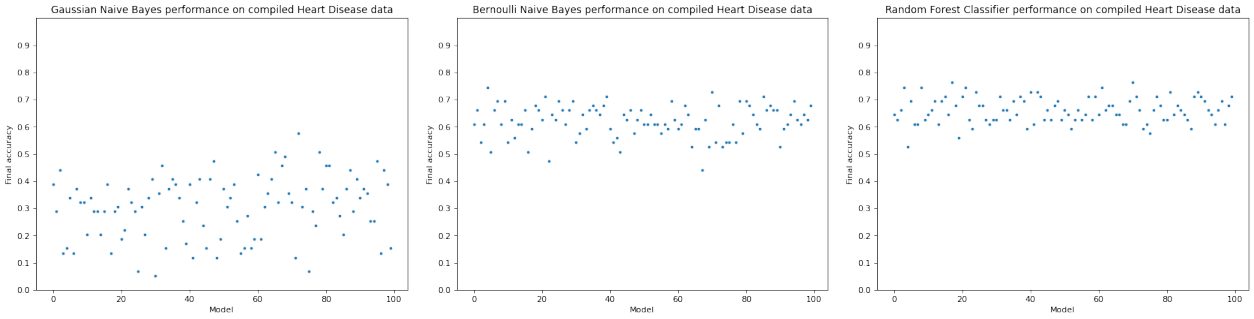
As seen in Fig 3, generating ANN models with a batch size of 1 and a different PRNG per model on the un-normalized PID data set results in no ANN model reaching 100% accuracy. The best ANN model has an accuracy score of 80.56%, whereas the worst ANN model has an accuracy score of 55.56%, making for a difference of 25% accuracy points between the best and worst ANN models. Seeing that normalizing the PID data set significantly improves the performance of ANN models, we leave exploring the effects of normalizing different data sets for other types of ML algorithms as a future work.

B. Experiment 2: Predicting heart disease with Naïve Bayes and Random Forest classifiers

A wide variety of ML algorithms have been evaluated in their performance of predicting heart disease [26], [36], [37], and there is no consensus as to which algorithm consistently performs the best. Our second experiment follows a survey of three ML algorithms—a Gaussian Naïve Bayes, a Bernoulli Naïve Bayes, and a Random Forest classifer—for heart disease diagnosis. Our Python implementations of these algorithms use the Scikit-learn GaussianNaiveBayes, BernoulliNaiveBayes, and RandomForest classes and Python's default Mersenne Twister PRNG. We extend upon Bernando et al.'s [26] evaluation of these algorithms by introducing variation in the PRNG seeds. We first generate 100 models of each algorithm on the *Cleveland Heart Disease* data set, where each model is generated with a different PRNG seed. As per Bernando et al., each model uses an 80:20 train/test split.

As seen in Fig. 5a- 5c, the three ML algorithms observably produces a wide variety of models on *Cleveland Heart Disease* given different PRNG seeds. From our Gaussian Naïve Bayes algorithm, the best model has an accuracy score of 65.57%, and the worst has an accuracy score of 34.43%, making for a 31.15% difference in accuracy points. Additionally, the best Bernoulli Naïve Bayes model reaches an accuracy score of 67.21%, and the worst has an accuracy score of 40.98%, a difference of 26.23% accuracy points. And lastly, the Random Forest classifier produces its best model with an accuracy score of 72.13%, and its worst with an accuracy score of 42.63%, making for a difference of 29.51% accuracy points.

Although *Cleveland Heart Disease* is more frequently cited in publications [25], we extended our experiment to the remaining three data sets in the UCI Repository *Heart Disease* collection. Fig. 5d-5l show the results of generating 100



(a) (b) (c)

Fig. 4: 100 models generated from each of the Gaussian Naïve Bayes, Bernoulli Naïve Bayes, and Random Forest algorithms using unique PRNG seed values on a compiled data set containing the data from all four *Heart Disease* data sets.

Data set	# of instances	# of missing values	Maximum accuracy	Minimum accuracy	Range of accuracy	Mean accuracy	Standard dev (accuracy)
Cleveland Heart Disease	303	6	GNB: 65.57% BNB: 67.21% RF: 72.13%	GNB: 34.26% BNB: 40.98% RF: 42.62%	GNB: 31.15% BNB: 26.23% RF: 29.51%	GNB: 53.44% BNB: 54.54% RF: 56.98%	GNB: 6.21% BNB: 5.61% RF: 5.61%
Switzerland Heart Disease	123	273	GNB: 36% BNB: 56% RF: 60%	GNB: 0% BNB: 20% RF: 16%	GNB: 36% BNB: 36% RF: 44%	GNB: 17.08% BNB: 36.44% RF: 38.68%	GNB: 7.46% BNB: 7.4% RF: 9%
Hungary Heart Disease	294	0	GNB: 57.63% BNB: 74.58% RF: 84.75%	GNB: 10.17% BNB: 50.85% RF: 50.85%	GNB: 47.46% BNB: 23.73% RF: 33.9%	GNB: 31.25% BNB: 62.37% RF: 66.32%	GNB: 10.78% BNB: 4.98% RF: 6.87%
Long Beach Heart Disease	200	698	GNB: 30% BNB: 32.5% RF: 47.5%	GNB: 2.5% BNB: 2.5% RF: 20%	GNB: 27.5% BNB: 30% RF: 27.5%	GNB: 15.9% BNB: 15.7% RF: 33.35%	GNB: 5.18% BNB: 5.75% RF: 6.54%
Heart Disease (compilation)	920	977	GNB: 57.62% BNB: 74.58% RF: 76.27%	GNB: 5.08% BNB: 44.07% RF: 52.54%	GNB: 52.54% BNB: 30.51% RF: 23.73%	GNB: 30.37% BNB: 62.03% RF: 66.14%	GNB: 11.34% BNB: 5.92% RF: 4.72%

TABLE I: Summary of results from Gaussian Naïve Bayes (GNB), Bayesian Naïve Bayes (BNB), and Random Forest (RF) classifiers on *Heart Disease* data sets.

models from each algorithm on each of the *Switzerland Heart Disease*, *Hungary Heart Disease*, and *Long Beach Heart Disease* data sets. We observe that all three algorithms obtained higher accuracy scores on the *Cleveland* and *Hungary Heart Disease* data sets, but have lower scores on the *Switzerland* and *Long Beach Heart Disease* data sets. This display of inconsistency across data sets makes it difficult to determine whether the respective ML algorithms are indeed capable of predicting heart disease given the relevant features. To address this lingering uncertainty, we tested each algorithm on a singular data set compiling all four data sets in the *Heart Disease* database. A summary of the performances of each algorithm on each *Heart Disease* data set can be seen in Table I.

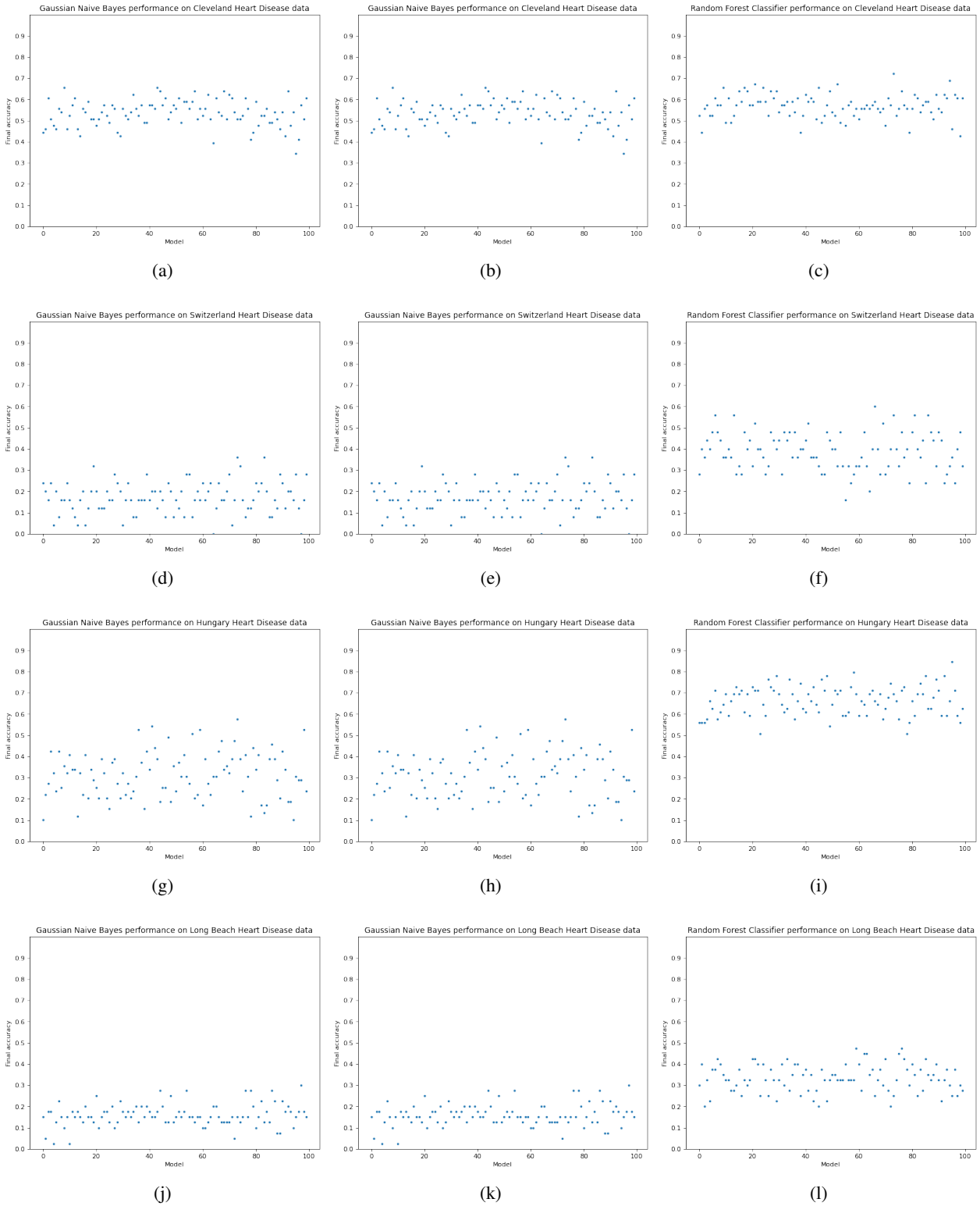
V. Discussion

Experiment 1 demonstrates both the variance of ANN model accuracy for diabetes prediction, and the difficulties of reproducing a published scientific result. This struggle to reproduce published work is experienced by many scientists, and is exactly the kind of problem that FAIR guidelines aim to rectify. Thus, Experiment 1 shows the need for FAIR standards and practices throughout the scientific research and publication process.

Experiment 1 also demonstrates that choice in data preprocessing steps and in training batch size are decisive of final model accuracy scores. Our work suggests that online training significantly increases the final accuracy scores of ANN models. Similar results were found by Won et al. [38], and we leave further investigation of our results as a future work.

Experiment 2 additionally quantifies the variance of Guassian Naïve Bayes, Bernoulli Naïve Bayes, and Random Forest classifers in medical diagnosis from numerous heart disease data sets. This experiment shows that comparing algorithmic performance on a given data set is incredibly difficult due to frequent inconsistency in model performance caused by PRNG seeds. We did observe that overall, the Random Forest algorithm had less variance and better testing accuracies on all four *Heart Disease* data sets separately (see Table I), although it only sometimes outperformed the Gaussian and Bernoulli Naïve Bayes algorithms on the compiled *Heart Disease* data set. We will leave further exploration of the potential for Random Forest classifiers to predict heart disease using different data sets as a future work.

Experiment 2 also showed that model accuracy can vary widely for different subsets of a data. We noted that in the *Heart Disease* data set collection, the number of missing



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 5: 100 models generated from each of the Gaussian Naïve Bayes, Bernoulli Naïve Bayes, and Random Forest algorithms on each of the *Heart Disease* data sets, where each model uses a unique PRNG seed values.

values varies by data set (see Table I). The *Swtizerland* and *Long Beach Heart Disease* data sets contain higher proportions of missing values, which likely contribute to all three algorithm's consistently lower performance on these respective data sets. This is relevant to our current work, in which we are investigating the factors that determine whether a given algorithm/data set pair will produce high or low accuracy models.

VI. Conclusions and Future Works

In this study, we described the obstacles in reproducing published scientific experiments, and quantified the variance in accuracy of common ML models (Artificial Neural Networks, Naïve Bayes classifiers, and Random Forest classifiers) in predicting different diseases. The data sets and code used for our experiments and data analysis can be found on GitHub.[1] The findings of this study determine that it can be near impossible to replicate another researcher's findings without identical raw data, programming code, and tooling. It is also difficult to verify the results of a comparison between ML algorithms on a given data set or task due to the extent of inconsistency caused by randomness.

As stated previously, current work involves determining the metrics that can be used to foretell whether a certain algorithm/data set pair will results in high quality models. Future work includes designing Python intercepts (similar to the C++ intercepts introduced by Ahmed et al. [22]) for capturing PRNG seeds and replacing the default Mersenne Twister in the Python `random` library with a Philox generator for better reproducibility across hardware systems. Another future work to this paper is running identical experiments with additional types of algorithms for prediction from medical data sets, and using a broader variety of hardware and high-performance computing tools such as GPUs and parallel programs.

Acknowledgment

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

[1] E. National Academies of Sciences, Medicine *et al.*, "Reproducibility and replicability in science," 2019.

[2] T. Griffin, Y. Cao, B. Liu, and M. J. Brunette, "Object detection and segmentation in chest x-rays for tuberculosis screening," in *2020 Second International Conference on Transdisciplinary AI (TransAI)*, 2020, pp. 34–42.

[3] T. Griffin, Q. Chen, X. Sun, D. Wang, M. J. Brunette, Y. Cao, and B. Liu, "erxnet: A pipeline of convolutional neural networks for tuberculosis screening," in *2021 Third International Conference on Transdisciplinary AI (TransAI)*, 2021, pp. 47–56.

[4] B. X. B. Yu, Y. Liu, and K. C. C. Chan, "Skeleton-based detection of abnormalities in human actions using graph convolutional networks," in *2020 Second International Conference on Transdisciplinary AI (TransAI)*, 2020, pp. 131–137.

[1] https://github.com/hahmed17/managing-randomness-for-medical-diagnosis

[5] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[6] M. Baker, "Reproducibility crisis," *Nature*, vol. 533, no. 26, pp. 353–66, 2016.

[7] J. P. Ioannidis, "Why most published research findings are false," *PLoS medicine*, vol. 2, no. 8, p. e124, 2005.

[8] M. Almada, L. Midão, D. Portela, I. Dias, F. Núñez-Benjumea, C. L. Parra-Calderón, and E. Costa, "A new paradigm in health research: Fair data (findable, accessible, interoperable, reusable)," 2020.

[9] E. T. Inau, J. Sack, D. Waltemath, and A. A. Zeleke, "Initiatives, concepts, and implementation practices of fair (findable, accessible, interoperable, and reusable) data principles in health data stewardship practice: Protocol for a scoping review," *JMIR Res Protoc*, vol. 10, no. 2, p. e22505, Feb 2021. [Online]. Available: https://www.researchprotocols.org/2021/2/e22505

[10] ——, "Initiatives, concepts, and implementation practices of fair (findable, accessible, interoperable, and reusable) data principles in health data stewardship practice: protocol for a scoping review," *JMIR research protocols*, vol. 10, no. 2, p. e22505, 2021.

[11] M. Basereh, A. Caputo, and R. Brennan, "Fair ontologies for transparent and accountable ai: A hospital adverse incidents vocabulary case study," in *2021 Third International Conference on Transdisciplinary AI (TransAI)*, 2021, pp. 92–97.

[12] S. Hooker, "Moving beyond "algorithmic bias is a data problem"," *Patterns*, vol. 2, no. 4, p. 100241, 2021.

[13] G. Yona, A. Ghorbani, and J. Zou, "Who's responsible? jointly quantifying the contribution of the learning algorithm and data," in *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 2021, pp. 1034–1041.

[14] R. E. Carter, Z. I. Attia, F. Lopez-Jimenez, and P. A. Friedman, "Pragmatic considerations for fostering reproducible research in artificial intelligence," *NPJ digital medicine*, vol. 2, no. 1, pp. 1–3, 2019.

[15] D. Zhuang, X. Zhang, S. Song, and S. Hooker, "Randomness in neural network training: Characterizing the impact of tooling," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 316–336, 2022.

[16] J. K. Salmon, M. A. Moraes, R. O. Dror, and D. E. Shaw, "Parallel random numbers: as easy as 1, 2, 3," in *Proceedings of 2011 international conference for high performance computing, networking, storage and analysis*, 2011, pp. 1–12.

[17] S. Scardapane and D. Wang, "Randomness in neural networks: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 2, p. e1200, 2017.

[18] H. V. Pham, S. Qian, J. Wang, T. Lutellier, J. Rosenthal, L. Tan, Y. Yu, and N. Nagappan, "Problems and opportunities in training deep learning software systems: An analysis of variance," in *Proceedings of the 35th IEEE/ACM international conference on automated software engineering*, 2020, pp. 771–783.

[19] S. S. Alahmari, D. B. Goldgof, P. R. Mouton, and L. O. Hall, "Challenges for the repeatability of deep learning models," *IEEE Access*, vol. 8, pp. 211 860–211 868, 2020.

[20] K. Sen, "Race directed random testing of concurrent programs," in *Proceedings of the 29th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2008, pp. 11–21.

[21] P. Frederickson, R. Hiromoto, and J. Larson, "A parallel monte carlo transport algorithm using a pseudo-random tree to guarantee reproducibility," *Parallel Computing*, vol. 4, no. 3, pp. 281–290, 1987.

[22] H. Ahmed and J. Lofstead, "Managing randomness to enable reproducible machine learning," in *Proceedings of the 5th International Workshop on Practical Reproducible Evaluation of Computer Systems*, 2022, pp. 15–20.

[23] C. Bogardus and S. Lillioja, "Pima indians as a model to study the genetics of niddm," *Journal of cellular biochemistry*, vol. 48, no. 4, pp. 337–343, 1992.

[24] J. J. Khanam and S. Y. Foo, "A comparison of machine learning algorithms for diabetes prediction," *ICT Express*, vol. 7, no. 4, pp. 432–439, 2021.

[25] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[26] C. Bernando, E. Miranda, and M. Aryuni, "Machine-learning-based prediction models of coronary heart disease using naïve bayes and random forest algorithms," in *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*. IEEE, 2021, pp. 232–237.

[27] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.

[28] N. S. El_Jerjawi and S. S. Abu-Naser, "Diabetes prediction using artificial neural network," *International Journal of Advanced Science and Technology*, vol. 121, 2018.

[29] N. Pradhan, G. Rani, V. S. Dhaka, and R. C. Poonia, "Diabetes prediction using artificial neural network," in *Deep Learning Techniques for Biomedical and Health Informatics*. Elsevier, 2020, pp. 327–339.

[30] P. M. Radiuk, "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets," 2017.

[31] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," *arXiv preprint arXiv:1804.07612*, 2018.

[32] L. C. Jain, M. Seera, C. P. Lim, and P. Balasubramaniam, "A review of online learning in supervised neural networks," *Neural computing and applications*, vol. 25, no. 3, pp. 491–509, 2014.

[33] D. R. Wilson and T. R. Martinez, "The general inefficiency of batch training for gradient descent learning," *Neural networks*, vol. 16, no. 10, pp. 1429–1451, 2003.

[34] J. Lin and D.-X. Zhou, "Online learning algorithms can converge comparably fast as batch learning," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2367–2378, 2017.

[35] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Transactions on nuclear science*, vol. 44, no. 3, pp. 1464–1468, 1997.

[36] E. O. Olaniyi, O. K. Oyedotun, A. Helwan, and K. Adnan, "Neural network diagnosis of heart disease," in *2015 International Conference on Advances in Biomedical Engineering (ICABME)*. IEEE, 2015, pp. 21–24.

[37] A. Rajdhan, A. Agarwal, M. Sai, D. Ravi, and P. Ghuli, "Heart disease prediction using machine learning," *International Journal of Research and Technology*, vol. 9, no. 04, pp. 659–662, 2020.

[38] J.-Y. Won, X. Chen, P. Gratz, J. Hu, and V. Soteriou, "Up by their bootstraps: Online learning in artificial neural networks for cmp uncore power management," in *2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2014, pp. 308–319.