Computer Methods and Programs in Biomedicine 213 (2022) 106504

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models

Alexandre Bailly[a,b,*], Corentin Blanc[a,b], Élie Francis[a], Thierry Guillotin[a], Fadi Jamal[c], Béchara Wakim[d], Pascal Roy[b]

[a] Everteam Software, Research and Development Lab, 17 quai Joseph Gillet, Lyon, France
[b] Université de Lyon, Lyon, France; Université Lyon 1, Villeurbanne, France; Service de Biostatistique-Bioinformatique, Hospices Civils de Lyon, Lyon, France; Équipe Biostatistique-Santé, Laboratoire de Biométrie et Biologie Évolutive, CNRS UMR 5558 Villeurbanne, France
[c] izyCardio - CardioParc, Lyon, France
[d] Mediapps Innovation SA, Lyon, France

ARTICLE INFO

Article history:
Received 1 April 2021
Accepted 24 October 2021

Keywords:
Deep learning
Machine learning
Data set size
Interactions

ABSTRACT

Background and objective: Machine learning and deep learning models are very powerful in predicting the presence of a disease. To achieve good predictions, those models require a certain amount of data to train on, whereas this amount i) is generally limited and difficult to obtain; and, ii) increases with the complexity of the interactions between the outcome (disease presence) and the model variables. This study compares the ways training dataset size and interactions affect the performance of those prediction models.

Methods: To compare the two influences, several datasets were simulated that differed in the number of observations and the complexity of the interactions between the variables and the outcome. A few logistic regressions and neural networks were trained on the simulated datasets and their performance evaluated by cross-validation and compared using accuracy, F1 score, and AUC metrics.

Results: Models trained on simulated datasets without interactions provided good results: AUC close to 0.80 with either logistic regression or neural networks. Models trained on simulated dataset with order 2 interactions led also to AUCs close to 0.80 with either logistic regression or neural networks. Models trained on simulated datasets with order 4 interactions led to AUC close to 0.80 with neural networks and 0.85 with penalized logistic regressions. Whatever the interaction order, increasing the dataset size did not significantly affect model performance, especially that of machine learning models.

Conclusion: Machine learning models were the less influenced by the dataset size but needed interaction terms to achieve good performance, whereas deep learning models could achieve good performance without interaction terms. Conclusively, with the considered scenarios, well-specified machine learning models outperformed deep learning models.

1. Introduction

In the last decades, Artificial Intelligence (AI) was largely used in a wide range of fields such as Pattern Recognition [1], Image Recognition [2], or Natural Language Processing [3]. In medicine, the main challenges have been assessing a disease risk, establishing a diagnosis, making a prognosis, or predicting the response to a treatment. All these challenges have been separately or jointly investigated in numerous medical specialties. Nevertheless, the superiority of AI methods over others was not always conclusive because of peculiarities of some methods, some diseases, some dataset contents or structure and, especially in prediction studies, because of the variety of metrics used to evaluate and compare model predictive performance; e.g., accuracy, sensitivity, specificity, area under the receiver operating curve (AUC), F1 score, etc.

Among the several hundreds of recent studies about the use of Deep Learning or Neural Networks for the analysis of medical data or, more specifically, prediction in medicine, a few selected works are briefly summarized herein.

* Corresponding author.
E-mail address: a.bailly@everteam.com (A. Bailly).

https://doi.org/10.1016/j.cmpb.2021.106504

In 2019, Ayon et al. [4] evaluated the performance of a deep neural network in predicting diabetes mellitus. With five-fold and ten-fold cross-validations on the Pima Indians Diabetes Dataset, the validations showed respectively 98.04 and 97.27% accuracy, 98.80 and 97.8% sensitivity, 96.64 and 96.27% specificity, 0.99 and 0.98 F1 score, and, finally, 0.96 and 0.94 Matthew's Correlation Coefficient. The deep neural network outperformed other current methods (e.g., with logistic regression, accuracy was only 78%).

In 2019 too, Tomita et al. [5] compared the performance of logistic regression, support vector machine, and deep neural network in making an initial diagnosis of adult asthma. The study included 566 adults with non-specific symptoms who visited a university hospital for the first time. Ten to 22 inputs were used with each model and performance assessed in terms of accuracy and AUC. With 10 symptom-physical signs as inputs, the accuracies were 65, 62, and 68%, respectively, but, with 22 inputs (10 + biochemical, functional, and other tests) the accuracies were 94, 82, and 98%, respectively, whereas the AUCs were 72.6, 54.5, and 63.2, respectively, with 10 inputs but 0.97, 0.83, and 0.99, respectively, with 22 inputs. Thus, with all inputs available, the neural network outperformed clearly the other methods in diagnosing adult asthma.

In 2020, Nazari et al. [6] used neural networks (with one and three hidden layers) and deep learning for the diagnosis of myeloid leukemia using microarray gene data from the Gene Expression Omnibus (GEO) database. The accuracies with a single-layer and three hidden layers were 63.33 and 96.67%, respectively, showing the power of the latter method.

In 2021, Lewis et al. [7] compared deep learning models with logistic regressions in predicting preventable acute care use and spending among 93,260 heart failure patients from a single large US insurer database. With all outputs predicted (preventable hospitalization, preventable ED visits and preventable costs), deep learning models showed the highest performance (e.g., concerning preventable hospitalization, the AUC was 0.75 for logistic regression vs. 0.77 for non-sequential neural network and 0.78 for sequential neural networks).

Machine Learning (ML) is a branch of AI based on learning from data using variables, also called 'features', to predict an 'outcome' (disease risk, diagnosis, prognosis, or response to treatment). Most ML models are trained with supervised learning, which implies a known outcome. Several models based on data training do exist (e.g., Linear Regression [8], Logistic Regression [9], Support Vector Machines [10], or Naive Bayes classifier [11]) in which it has been shown that increasing the amount of data improves performance [12]. In the field of classification, performance is the ability of a model to predict an observation's class using a test dataset. A bad performance of a classifier may have two causes: i) the features do not contain enough information to explain the outcome; and, ii) the model cannot deal with the complexity of the relationships between the features and the outcome. Generalized Linear Models are able to unravel part of this complexity by introducing interaction terms of various orders, complexity being considered as a deviation from an additive effect between the model's linear components. Unfortunately, in case of high complexity, ML has to consider high-order interaction terms during model building.

Deep Learning (DL) is a branch of ML that includes models with more elaborated architectures. The simplest DL model is the perceptron [13], an interconnection of artificial neurons. The perceptron may be extended by adding hidden layers to form a deeper network named 'multilayer perceptron'. A onelayer perceptron is able to approximate any continuous function; this was proven by Hornik et al. [14,15] with the Universal Approximation Theorem. One implication of that Theorem is that, in a fully connected NN, each neuron after the first one in the first hidden layer takes into account all interactions between the features. With a one-layer perceptron, DL models are thus able to deal with relationship complexity by including high-order interaction terms, especially that they do not have to be specified. However, DL models require large training datasets [16].

Some metrics were developed to evaluate the predictive performance of mod- els; e.g., accuracy, AUC, or F1 score. These metrics allow model comparisons of very different types (e.g., a ML model vs. a DL model). Concerning ML approaches, studies such as those of Tsangaratos et al. [12] compared model performance by varying both the number of observations and the number of features used in the models with real drug datasets. However, these studies did not deal with DL models. Interestingly, Korotcov et al. [17] compared the performance of DL and ML models in terms of accuracy and AUC. With all datasets used, DL models gave better results than ML models but the authors did not systematically explore various levels of complexity. Consequently, those results were specific to the datasets used and still have be confirmed. Van der Ploeg et al. [18] created simulated datasets from different cohorts to study the amount of data needed by Logistic Regression or DL models to predict binary outcomes. They showed that Logistic Regression models needed less data than DL models and suggested DL models be used only with large datasets. Overall, the latter works took into account the amount of data but not the interaction orders. Theoretical studies evaluating the abilities of various models to deal with high interaction orders and their effects on performance are still needed.

The aim of the present study was to compare the prediction performance of ML models vs. that of DL models according to the training dataset size and the complexity of the interactions between the variables and the outcome. The variables and the presence of disease were simulated in virtual patients and complexity was generated by introducing multiplicative interaction terms in the models.

This report presents first the data, the way variables and subject statuses were simulated, the way the scenarios were built, the models used for prediction, and then the criteria used for comparisons of models' predicting abilities. The results under various scenarios are then displayed before being discussed and finally summarized in a clear conclusion.

2. Materials and methods

2.1. Simulated data

The Framingham Study [19] is one of the longest and most important epidemiological studies in medical history. It was a population-based observational cohort study initiated in 1948 to investigate prospectively the epidemiology and risk factors for cardiovascular disease in nearly 5210 participants, all residents of Framingham (MA, USA). Among various findings, that remarkable study demonstrated the detrimental roles of cigarette smoking, elevated blood pressure, and high cholesterol level in the development of heart disease and their contributions to the risk of heart attack and stroke. The original and subsequent databases include (but are not limited to) information on hundreds of demographic, clinical, laboratory, and imaging variables. (For more details see: Boston Medical Center. Framingham Study. https://www.bmc.org/stroke-and-cerebrovascular-center/research/framingham-study).

The present work adopted a short list of risk factors identified by the Framingham Study and simulated corresponding data with various levels of interaction to simulate outcomes. These risk factors were the following: age (in years), total cholesterol (in mg/dL), HDL cholesterol (in mg/dL), systolic blood pressure (in mm Hg), treatment for hypertension, smoking status, and diabetic status (as binary variables) (Table 1).

Table 1
Description of the features considered.

Features	Mean	Standard deviation	Proportion
Continuous features			
Age	50	5.92	–
Total cholesterol	215	4.47	–
HDL cholesterol	45	5.00	–
Systolic blood pressure	130	12.25	–
Binary features			
Treatment for hypertension	–	–	0.1013
Smoking status	–	–	0.3522
Diabetes status	–	–	0.0650

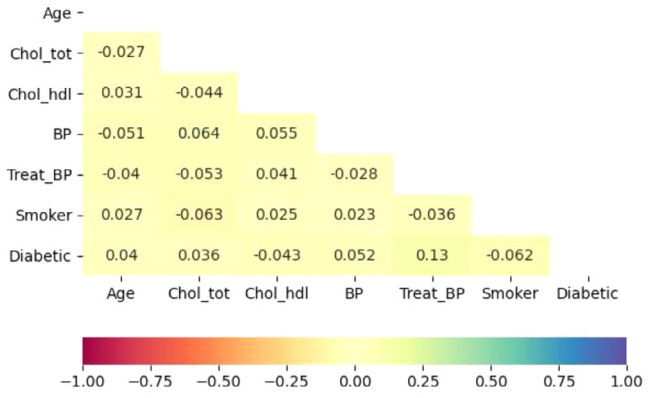

Fig. 1. Pairwise correlation between features.

2.1.1. Feature generation

Each continuous feature was randomly generated according to a Gaussian distribution having the same mean as in the Framingham study [19] and a standard deviation chosen according to common values seen in clinical practice (Table 1). Binary features were generated using binomial laws where each probability parameter corresponded to the distribution of the feature in the Framingham study (Table 1). All features were independently generated.

Fig. 1 shows pairwise correlations between features. Because several datasets were simulated, only the highest correlations (in absolute value) are shown. As the maximum correlation value was 0.13, we considered that there is almost no correlation between the features. This stems from the variable generation process where all the features were independently simulated.

2.1.2. Clinical status generation

Clinical statuses were dependent on the features and their possible interactions. Here, only multiplicative interactions between features were considered. Considering $P(X)$ as a polynomial created from a vector of features X, the probability of being a 'case' was computed with the following formula:

$$p = \frac{1}{1 + e^{-P(X)}}$$

Once a probability was computed for a given observation, a binomial law having this probability as parameter was used to assign a clinical status (case or non-case). In this process, the order of the interaction is the degree of P used to assign the clinical status.

2.1.3. Scenario generation

Different datasets were generated with sizes 1,000, 10,000, and 100,000 observations. Furthermore, three polynomials were used to assign cases to observations (see the Appendix): i) only the main effects considered (the features without interaction terms); ii) main effects and interaction terms of order 2 considered; and, iii) main effects and interaction terms up to order 4 considered. In this study, interactions within a single feature were not considered. Nine scenarios were thus generated (three sizes x three polynomials).

2.2. Models

2.2.1. Machine learning models

The first ML model was logistic regression (LR) [9]. LR is derived from the Linear Model where the outcome variable is binary. In a LR with k features, the log of the odd of the probability p is modeled from features $X = (x_1,\bullet\bullet\bullet,x_k)$ by a linear model:

$$log\left(\frac{p}{1-p}\right) = \sum_{j=1}^{k} \beta_j x_j$$

$$\Leftrightarrow \quad p = \frac{1}{1 + e^{-\sum_j \beta_j x_j}}$$

Parameters $\beta = (\beta_1,\bullet\bullet\bullet,\beta_k)$ were estimated by maximizing the log-likelihood function. With n observations, y_i and x_i being respectively the clinical status and the vector of features for the i-th observation, the log-likelihood function was given by:

$$l\beta \sum_{i=1}^{n} \left[y_i x_i \beta - log\left(1 + e^{x_i \beta}\right)\right]$$

Variable selection was made by comparing nested models using the likelihood ratio test. In a first step, the main effects were kept only when the associated p-values were < 0.05. In a second step, all order-2 interactions were successively tested by fitting models that included systematically all previously retained main effects. For interaction tests, p-values < 0.10 were considered. Furthermore, a model that included the previously retained main effects and all identified order-2 interactions was fitted; this led to consider only order-2 interactions with p-value still < 0.10. Starting from that model, order-3 then order-4 interaction terms were selected using similar approaches.

Two types of penalization were applied: Lasso and Ridge. The function to maximize was then:

$$l_\lambda^\omega(\beta) = \sum_{i=1}^{n} \left[y_i x_i \beta - \log\left(1 + e^{x_i \beta}\right)\right] - \lambda \sum_{j=1}^{k} \left|\beta_j\right|^\omega$$

In this formula, λ is the tuning parameter and $\omega = 1$ for Lasso or 2 for Ridge. In both situations, penalization led to shrinkage of parameters β. With Lasso penalization, a selection may be carried out by setting β to 0.

With each penalization method, a model that included all main effects was fitted first then a model that included all main effects and all order-2 interaction terms. Finally, a model that included all main effects and all order-2, order-3, and order-4 interaction terms was fitted. The tuning parameter was fixed to 1.

Hereafter, notations LR_i, $lasso_i$, and $ridge_i$ will indicate the different logistic regressions trained with the variables and their interactions up to order i.

2.2.2. Deep learning models

Neural networks (NNs) [1], just like a multilayer perceptron, are based on artificial neurons. These neurons are connected in many layers to create a network. The input layer includes as many neurons as there are features and the output layer as many neurons as there are output possibilities. Between these two layers, some other hidden layers may be included. Here, a sigmoid function was used as activation function for all layers, except the last one for which log-softmax function was used. In addition, between hidden layers, a dropout with probability 0.1 was considered to reduce overfitting. All NNs were trained by backpropagation [20] to

Table 2
Metrics relative to the main effects.

	Dataset size: 1000			Dataset size: 10000			Dataset size: 100000		
Model	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC
LR_1	0.70	0.48	0.70	0.71	0.49	0.71	0.71	0.49	0.71
LR_2	0.76	0.54	0.74	0.78	0.58	0.78	0.79	0.58	0.78
LR_4	0.76	0.55	0.75	0.79	0.59	0.78	0.80	0.60	0.79
$lasso_1$	0.89	0.69	0.80	0.89	0.70	0.80	0.89	0.69	0.79
$lasso_2$	0.89	0.69	0.79	0.89	0.70	0.80	0.89	0.69	0.79
$lasso_4$	0.88	0.68	0.78	0.89	0.70	0.80	0.89	0.69	0.79
$ridge_1$	0.89	0.70	0.80	0.89	0.70	0.80	0.89	0.69	0.79
$ridge_2$	0.88	0.68	0.79	0.89	0.69	0.80	0.89	0.69	0.79
$ridge_4$	0.88	0.67	0.78	0.89	0.69	0.79	0.89	0.69	0.79
NN_1	0.88	0.67*	0.79	0.89	0.68	0.78	0.89	0.67	0.78
NN_3	0.88	0.66†	0.77	0.88	0.66	0.77	0.89	0.68	0.78
NN_5	0.88	0.67†	0.79	0.88	0.65	0.77	0.88	0.67	0.78
NN_7	0.86	0.63‡	0.77*	0.88	0.65	0.77	0.88	0.66	0.77

†The SD for this value was 0.04 -
* the SD for this value was 0.05 -
‡ The SD for this value was 0.08 -All other SDs were ≤ 0.03

minimize the cross-entropy loss function [21] given, in a binary context, by:

$$L(y, p) = -(ylog(p) + (1 - y)log(1 - p))$$

In this equation, p is the probability returned by the model and y the clinical status (1 for 'case', 0 for 'non-case'). For the backpropagation, Adam optimizer was used to update the parameters of the model [22]. Furthermore, to optimize the backpropagation process, the weights of the NNs were initialized using Xavier normal function, which is recommended for NNs with sigmoid activation functions [23]. The training process needed some other hyperparameters that were selected from sets of possible values using cross-validation. Doing so, four hyperparameters were fixed for each model in each scenario: i) the number of epochs (25, 50, or 100); ii) the batch size (32 or 64); iii) the learning rate used for backpropagation (0.01 or 0.1); and, iv) the number of neurons in each hidden layer (3, 5, or 7). Here, for simplicity, all layers had the same number of neurons and four different NNs with 1, 3, 5, or 7 hidden layers were considered. Hereafter, NN_i will denote a NN with i hidden layers.

2.3. Performance evaluation

2.3.1. Metrics

To evaluate the predictive performance of each model, three metrics were considered: i) accuracy; i.e., the proportion of well-classified observations; ii) the AUC; i.e., the probability of ranking a random positive observation higher than a random negative observation; and, iii) F1 score, a harmonic mean between precision and recall; i.e., between the proportion of positive identifications actually correct = TP/(TP+FP) and the proportion of actual positives correctly identified = TP/(TP+FN) (T, F, P, and N stand respectively for True, False, Positives, and Negatives).

2.3.2. Evaluation of the results

Ten-fold cross-validations were used to simulate an internal validation. Theoretically, to avoid random sample fluctuations inherent to dataset generation, a large number of datasets in each scenario should be simulated with their corresponding outcomes. Here, only five datasets were used to reduce computing times and because the observed fluctuations were very small. The results show, in each scenario, the mean value of each metric as computed by cross-validation on the five datasets.

3. Results

3.1. Main effects

In scenarios with only main effects used to assign cases, LR results were similar whatever the order of the interaction terms used for training. Thus, with 1,000 observations, LR1 had an AUC of 0.70, whereas LR_2 and LR_4 had AUCs of 0.74 and 0.75, respectively (Table 2). Table 2 shows also that using penalization (either Lasso or Ridge) led to an improvement of performance. Indeed, the AUCs were around 0.80 with Lasso and Ridge whatever the order of interaction terms used for training. Furthermore, Table 2 shows that increasing the dataset size did not improve performance whatever the penalization technique (AUC = 0.80 with 10,000 observations vs. 0.79 with 100,000 observations). The AUC was around 0.78 with non-penalized LRs, especially LR_2 and LR_4.

Concerning NNs, the AUCs were quite good: 0.79 (SD: 0.03) for NN_1, 0.77 (SD: 0.02) for NN_3, 0.79 (SD: 0.03) for NN_5, and 0.77) for NN_7 (Table 2). When only the main effects were considered, the number of hidden layers had only a limited effect on performance.

Table 2 shows that increasing the dataset size did not increase the values of the metrics; it only stabilized the AUC around value 0.77 for 10,000 observations and 0.78 for 100,000 observations.

Similar results were observed for accuracy and F1 score.

3.2. Interactions of order 2

When interactions of order 2 were used to assign cases, differences in performance appeared between different LRs. Table 3 shows that with 1,000 observations, the AUC was 0.71 for LR_1, 0.80 for LR_2, and 0.79 for LR_4. Penalization improved slightly the results of all classical LRs. Indeed, Table 3 shows that the AUCs with Lasso and Ridge were 0.75 when only the main effects were used for training, 0.82 otherwise. Increasing the dataset size improved slightly AUC values. Table 3 shows also AUC values of 0.71, 0.81, and 0.82 for LR_1, LR_2, and LR_4, respectively with dataset sizes $>$ 1,000. With dataset sizes 10,000 and 100,000, penalization improved the performances regardless of the order of interaction terms used for training: the AUC was 0.77 for ridge and lasso trained without interactions and 0.84 for ridge and lasso trained with interactions.

Concerning NNs, Table 3 shows similar performances whatever the number of hidden layers. With 1,000 observations, the AUC was 0.79 for NN_1, 0.81 for NN_3, 0.80 for NN_5, and 0.78 for NN_7.

Increasing the dataset size stabilized the AUC for all NNs but did not improve performance. Table 3 shows that the AUC was 0.81

Table 3
Metrics relative to order-2.

	Dataset size: 1000			Dataset size: 10000			Dataset size: 100000		
Model	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC
LR_1	0.72	0.50	0.71	0.72	0.49	0.71	0.72	0.49	0.70
LR_2	0.80	0.62	0.80	0.81	0.63	0.81	0.82	0.63	0.81
LR_4	0.80	0.61	0.79	0.83	0.65	0.82	0.84	0.66	0.82
$lasso_1$	0.88	0.64[$]	0.75 [†]	0.89	0.67	0.77	0.89	0.66	0.76
$lasso_2$	0.90	0.74	0.82	0.92	0.77	0.84	0.92	0.77	0.84
$lasso_4$	0.90	0.74	0.82	0.92	0.77	0.84	0.92	0.77	0.84
$ridge_1$	0.88	0.64[$]	0.76 [†]	0.89	0.67	0.77	0.89	0.65	0.76
$ridge_2$	0.91	0.74	0.82	0.92	0.77	0.84	0.92	0.77	0.84
$ridge_4$	0.90	0.73	0.82	0.92	0.77	0.84	0.92	0.77	0.84
NN_1	0.89	0.69 [†]	0.79	0.90	0.73	0.81	0.90	0.72	0.81
NN_3	0.89	0.71	0.81	0.90	0.73	0.81	0.91	0.73	0.81
NN_5	0.88	0.69[*]	0.80	0.90	0.73	0.81	0.90	0.72	0.81
NN_7	0.87	0.66[+]	0.89[*]	0.90	0.72	0.80	0.90	0.72	0.81

[†] The SD for this value was 0.04 -
[*] The SD for this value was 0.05
[$] The SD for this value was 0.06 -
[+] The SD for this value was 0.07 - All other SDs were $\leq$ 0.03

Table 4
Metrics relative to order-4 interactions.

	Dataset size: 1000			Dataset size: 10000			Dataset size: 100000		
Model	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC	Accuracy	F1-score	AUC
LR_1	0.73	0.49[*]	0.71	0.73	0.49	0.70	0.73	0.49	0.70
LR_2	0.80	0.59	0.78	0.81	0.61	0.79	0.81	0.61	0.79
LR_4	0.80	0.60	0.79	0.83	0.65	0.82	0.83	0.65	0.83
$lasso_1$	0.89	0.64[$]	0.76 [†]	0.89	0.66	0.76	0.89	0.66	0.76
$lasso_2$	0.90	0.71 [†]	0.80	0.90	073	0.81	0.91	0.73	0.81
$lasso_4$	0.91	0.74 [†]	0.82	0.92	0.78	0.85	0.92	0.79	0.85
$ridge_1$	0.89	0.64[$]	0.76	0.89	0.66	0.77	0.89	0.66	0.76
$ridge_2$	0.90	0.70 [†]	0.80	0.90	0.73	0.81	0.91	0.73	0.81
$ridge_4$	0.91	0.74	0.83	0.92	0.78	0.85	0.92	0.79	0.85
NN_1	0.89	0.66 [†]	0.77	0.90	0.71	0.80	0.90	0.71	0.80
NN_3	0.89	0.70	0.80	0.90	0.72	0.81	0.90	0.73	0.81
NN_5	0.88	0.67 [†]	0.79	0.90	0.72	0.81	0.90	0.72	0.81
NN_7	0.89	0.69	0.80	0.90	0.71	0.80	0.90	0.71	0.81

†The SD for this value was 0.04
[*] The SD for this value was 0.05
[$] The SD for this value was 0.06 - All other SDs were $\leq$ 0.03

for all NNs, except NN_7 with 10,000 observations (AUC = 0.80). Similar results were observed for accuracy and F1 score.

3.3. Interaction of order 4

When interactions of order 4 were used to assign cases, the performance of LR differed according to the order of interaction terms used during the training.

Table 4 shows that for 1,000 observations, the AUC was 0.71 for LR_1, 0.78 for LR_2, and 0.79 for LR_4. When the relationship between the features and the outcome required one or more interactions of order 4, penalization improved performance. Table 4 shows that the AUCs with Lasso and Ridge were above 0.76, 0.80, and 0.82 for penalized LR with main effects, main effects and order-2 interactions, main effects and all interactions up to order 4, respectively. Increasing the dataset size improved performance (Table 4). This improvement concerned especially penalized and non-penalized LRs with order-4 interaction terms. Indeed, the AUC for $lasso_4$ and $ridge_4$ peaked at 0.85 with 10,000 or 100,000 observations. Furthermore, the AUCs for LR4 were 0.82 and 0.83 for 10,000 and 100,000 observations, respectively. Nevertheless, for LRs without order-4 interaction, increasing the dataset size did not improve performance.

Concerning NNs, the number of hidden layers had no important impact on performance (Table 4). The AUCs were 0.77 for NN_1, 0.80 for NN_3, 0.79 for NN_5, and 0.80 for NN_7.

Table 4 shows also that increasing the dataset size stabilized the metric values rather than increasing them. Indeed, the AUC was around 0.80 for all NNs whatever the number of hidden layers.

Similar results were obtained for accuracy and F1 score.

4. Discussion

This work aimed at investigating the effects of one form of complexity (i.e., logistic interactions) and dataset size on logistic regression and neural network predicting abilities. The former effect was investigated with simulated data. Assuming that neural networks can approach any function in a complex space, low and high interaction orders were simulated. Another form (namely, model complexity) was used to compare the prediction ability of logistic regression (with various levels of interaction used as inputs and various penalization functions) vs. neural networks (with different numbers of hidden layers).

In all studied scenarios, penalization of the LR improved performance of all models. In addition, very few changes were seen by varying the size of the dataset, except for NNs. Unsurprisingly, LR with interaction terms provided better results than other models, especially with datasets with order-4 interactions. NNs provided good results with almost all datasets and had nearly the same level of performance than LR with datasets with interaction of order 1in scenarios without interaction terms. With higher interaction orders, NNs performed less well than LRs that

used the right interaction term orders. Despite this, when simulated data integrated interaction terms, NNs outperformed LRs that did not include such terms. With order-4 interactions, NNs provided similar results to LRs trained with main effects and order-2 interactions.

The present results show that almost all scenarios with Lasso or Ridge penalization improved performance. This was not surprising, especially with interactions of high orders. Indeed, in LRs, increasing the interaction order increases the number of variables used as inputs. Term selection during training classical LRs led to optimism (overfit with the training dataset leading to worse predictions with the test dataset) [24]. This problem may be solved using penalization techniques, either Lasso or Ridge. With penalization, the predictive performance improved but neither penalization method outperformed the other.

Concerning NNs, the performance was good but less than that of wellspecified LRs. One may recall that optimizing NNs is more complex than optimizing LRs. A high number of hyperparameters must be set to train NNs. This study used a restricted number of possible values for each hyperparameter.

More combinations of hyperparameters may lead to better optimization of NNs and, thereby, to better performance. Furthermore, training of NNs requires more computation time than training LRs.

Neural networks and logistic regression are subject to uncertainty. There are two types of uncertainty: aleatory uncertainty (the uncertainty inherent to the data and that the modeler cannot reduce) and epistemic uncertainty (the uncertainty due to the model design and that can be reduced by the modeler) through adding knowledge [25]. Though the effects of interaction terms on uncertainty were not studied yet, we expect that increasing the dataset size would decrease epistemic uncertainty (for a recent technical point on the techniques able to quantify uncertainty, see the comprehensive review of Abdar et al. [26]).

Two peculiarities of the present study is that, by construction, the data were idealized; i.e., free from bias. The lack of bias may inflate NN performance in cases of large datasets. Besides, the data were homoscedastic; i.e., generated as if they were drawn from a single population. The latter fact might not be true in other real-world datasets or in multicenter datasets. In case of several data sources, these points should be taken into account during model building. Moreover, the features were independently simulated (i.e., without any correlation) whereas correlations between features make interaction term influence model's prediction. Thus, correlations between features should be thoroughly examined. Furthermore, the effects of bias, heteroscedasticity, and correlation on prediction may be investigated in more extended studies.

One major asset of this study is the use of simulated data. Indeed, simulating the data allowed a better knowledge of their structure and complexity. In most cases, with real datasets, the complexity of the data structure is not known beforehand; consequently, its impact on models' performance is difficult (if not impossible) to assess and counterweight. In the present work, complexity was restricted to logistic interactions. Working with simulated data allows investigating the way NNs behave according to various forms and levels of complexity.

Consequently, a future work will examine the performance of LRs and NNs with datasets that display more complexity; e.g., nonlinear relationship between the features and the outcomes or interactions of very high orders. The abilities of other models such as Random Forest, Support Vector Machines, or Kernel regressions to approximate that relationship may also be evaluated considering various complexity levels. Besides, within the spirit of a previous review, a specific work might be dedicated to the effect of 'interaction' or 'dataset size' the two types of uncertainty [26].

5. Conclusions

The present study investigated the ability of LRs and NNs to deal with multiplicative interactions. With all interaction orders, well-specified LRs provided the best results. Furthermore, penalized LRs outperformed regular LRs. NNs performed at least as well as LRs without the right interaction order. The study showed no significant impact of the dataset size. DL can be a powerful tool but well-specified classical ML approaches are likely to be more efficient in many biomedical applications.

Funding

This work was supported by Association Nationale de la Recherche et de la Technologie (ANRT) [grantnumber 2019/1373] and by Service de Biostatistique-Bioinformatique des Hospices Civils de Lyon.

Declaration of Competing Interest

The authors have no competing interests to declare. Authors AB, CB, EF, and TG are employed by Everteam Software.

Acknowledgment

We would like to thank Jean Iwaz (Hospices Civils de Lyon) for his constructive feedback, useful comments, and valuable suggestions.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2021.106504.

References

[1] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Inc., USA, 1995.
[2] K.J. Cios, I. Shin, Image recognition neural network: IRNN, Neurocomputing 7 (1995) 159–185, doi:10.1016/0925-2312(93)E0062-I.
[3] D. Li, L. Yang, Deep Learning in Natural Language Processing, 1st Edition, Springer Publishing Company, 2018 Incorporated.
[4] S.I. Ayon, M.M. Islam, Diabetes prediction: a deep learning approach, Int. J. Inf. Eng. Electr. Bus. 11 (2019) 21–27, doi:10.5815/IJIEEB.2019.02.03.
[5] K. Tomita, R. Nagao, H. Touge, T. Ikeuchi, H. Sano, A. Yamasaki, Y. Tohda, Deep learning facilitates the diagnosis of adult asthma, Allergol. Int. 68 (2019) 456–461, doi:10.1016/j.alit.2019.04.010.
[6] E. Nazari, A.H. Farzin, M. Aghemiri, A. Avan, M. Tara, H. Tabesh, Deep learning for acute myeloid leukemia diagnosis, J. Med. Life 13 (2020) 382, doi:10.25122/jml-2019-0090.
[7] M. Lewis, G. Elad, M. Beladev, G. Maor, K. Radinsky, D. Hermann, Y. Litani, T. Geller, J.M. Pines, N. l. Shapiro, J.F. Figueroa, Comparison of deep learning with traditional models to predict preventable acute care use and spending among heart failure patients, Sci. Rep. 11 (2021), doi:10.1038/s41598-020-80856-3.
[8] K.H. Zou, K. Tuncali, S.G. Silverman, Correlation and simple linear regression, Radiology 227 (2003) 617–628, doi:10.1148/radiol.2273011499.
[9] S. Sperandei, Understanding logistic regression analysis, Biochemia Medica 24 (2014) 12–18, doi:10.11613/bm.2014.003.
[10] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowl. Discovery 2 (1998) 121–167, doi:10.1023/A:1009715923555.
[11] I. Rish, et al., An empirical study of the naive bayes classifier, in: IJCAI 2001workshop on empirical methods in artificial intelligence, 3, 2001, pp. 41–46.
[12] P. Tsangaratos, I. Ilia, Comparison of a logistic regression and Naïve Bayesclassifier in landslide susceptibility assessments: The influence of models complexity and training dataset size, Catena 145 (2016) 164–179, doi:10.1016/j.catena.2016.06.004.
[13] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (1958) 386, doi:10.1037/h0042519.
[14] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366, doi:10.1016/0893-6080(89)90020-8.
[15] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Netw. 4 (1991) 251–257, doi:10.1016/0893-6080(91)90009-T.

[16] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M.M.A. Patwary, Y. Yang, Y. Zhou, Deep Learning Scaling is Predictable, Empirically (2017) arXiv:1712.00409 [cs, stat].
[17] A. Korotcov, V. Tkachenko, D.P. Russo, S. Ekins, Comparison of Deep Learning With Multiple Machine Learning Methods and Metrics Using Diverse Drug Discovery Data Sets, Mol. Pharm. 14 (2017) 4462–4475, doi:10.1021/acs.molpharmaceut.7b00578.
[18] T. van der Ploeg, P.C. Austin, E.W. Steyerberg, Modern modelling techniquesare data hungry: a simulation study for predicting dichotomous endpoints, BMCMed. Res. Methodol. 14 (2014), doi:10.1186/1471-2288-14-137.
[19] R.B. D'Agostino, S.V. Ramachandran, M.J. Pencina, P.A. Wolf, M. Cobain, J.M. Massaro, W.B. Kannel, General Cardiovascular Risk Profile for Use in Primary Care, Circulation 117 (2008) 743–753, doi:10.1161/circulationaha.107.699579.
[20] Y.A. LeCun, L. Bottou, G.B. Orr, K.-R. M üller, Efficient BackProp, in: G. Montavon, G.B. Orr, K.R.M. Üller (Eds.), Neural Networks: Tricks ofthe Trade, Second Edition, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 9–48, doi:10.1007/978-3-642-35289-8_3.
[21] Z. Zhang, M. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, in: Advances in neural information processing systems, 2018, pp. 8778–8788.
[22] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, arXiv:1412.6980 [cs] (2017).
[23] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y.W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Vol. 9 of Proceedings of Machine Learning Research, PMLR, Sardinia, Italy, Chia Laguna Resort, 2010, pp. 249–256.
[24] Y. Zhao, E. Dantony, P. Roy, Optimism Bias Correction in Omics Studies with Big Data: Assessment of Penalized Methods on Simulated Data, OMICS 23 (2019) 207–213, doi:10.1089/omi.2018.0191.
[25] A.D. Kiureghian, O. Ditlevsen, Aleatory or epistemic? Does it matter? Struct. Saf. 31 (2009) 105–112, doi:10.1016/j.strusafe.2008.06.020.
[26] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P.W. Fieguth, X. Cao, A. Khosravi, U.R. Acharya, V. Makarenkov, S. Nahavandi, A review of uncertainty quantification in deep learning: Techniques, applications and challenges, CoRR abs/2011.06225 (2020). https://doi.org/10.1016/j.inffus.2021.05.008.