DOI: 10.24996/ijs.2025.66.9.28

ISSN: 0067-2904

Performance Evaluation of Some Machine Learning Regression Models with Application

Asmaa Ali Zaidan[1*] , Tasnim Hasan Kadhim[2]

[1]*Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq.*

[2]*Assistant Professor, Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq.*

Received: 27/1/2024 Accepted: 9/9/2024 Published: 30/9/2025

Abstract:

Currently, Machine learning is an advanced algorithm that yielding accurate classifications or predictions for huge samples sizes. The fat index data is asymmetric and has a right skew so this creates problem when using statistical techniques. For such data, there is still a lot of performance that needs to be improved when comparing statistical techniques and machine learning algorithms. This paper aimed to compare the traditional statistical methods represented by Linear, penalized linear such as Ridge and Lasso regression and machine learning models represented by convolutional neural network for their prediction performance through simulation experiments and real data of fat index. A total of 252 records were used. The prediction performance of fat index by Linear, Ridge, Lasso and Convolutional Network were compared using mean square error and mean absolute error. We concluded that Ridge and linear regression had the worst performance with the biggest mean and absolute prediction errors, while the Convolutional Neural Network technique achieved the lowest mean and absolute prediction errors, providing the best predictive performance for the fat index data and simulation experiments.

Keywords: Linear Model, Lasso Method, Ridge Regression, Convolutional Neural Network, Machine Learning.

تقييم أداء بعض نماذج انحدار التعلم الالي مع تطبيق

اسماء علي زيدان[1]*, تسنيم حسن كاظم[2]

[1]قسم الرياضيات, كلية العلوم,جامعة بغداد,بغداد, العراق

[2] أستاذ مساعد، قسم الرياضيات, كلية العلوم,جامعة بغداد,بغداد, العراق

الخلاصة

التعلم الآلي او تعلم الماكنة عبارة عن خوارزمية متقدمة توفر تصنيفات أو تنبؤات دقيقة عند استخدام حجوم العينات الضخمة. تظهر بيانات مؤشر الدهون او البدانة بشكل عام توزيع ملتوي لليمين وغير متماثل وهذا يسبب مشكلة عند استخدام التقنيات الإحصائية التقليدية. بالنسبة لمثل هذه البيانات، لا يزال هناك الكثير من المجالات التي تستدعي المقارنة التقنيات الإحصائية وخوارزميات التعلم الآلي لغرض تحسين نمذجة تلك

*Email: asmaa.ali2203m@sc.uobaghdad.edu.iq

البيانات. يهدف هذا البحث إلى مقارنة الأساليب الإحصائية التقليدية المتمثلة بالنماذج الخطية ، نماذج التقلص وانحدار لاسو مع خوارزميات التعلم الآلي المتمثلة بالشبكة العصبونية الالتفافية لغرض تحسين دقة تنبؤات مؤشر الدهون. تم استخدام بيانات بحجم عينة 252 مشاهدة. وتمت مقارنة أداء التنبؤ على مؤشر الدهون بواسطة تلك الطرائق واستخدام متوسط مربع الخطأ ومتوسط الخطأ المطلق. استنتج أن انحدار التقلص كان له أسوأ أداء بسبب القيمة الكبيرة لمتوسط مربعات الخطا التربيعي والمطلق ، في حين حققت تقنية الشبكة العصبونية الالتفافية أقل خطاء تنبؤي مطلق وتربيعي مما لها أفضل أداء لنمذجة بيانات مؤشر الدهون وتجارب المحاكاة.

1. Introduction:

Recently, deep learning, an area of machine learning has made an interesting application and success in many fields, ranges from on line speech translation to self-driving cars to real time. Machine learning shares applications and innovations with many fields such as mathematics, statistics, bioinformatics, medicine … etc.

Traditional Statistical methods which proved as rigorous methods in data analysis for many years require many assumptions such as: normality, linearity, clean data … etc. The quality of those methods depends on fulfilled of those assumptions.

Many researchers studied the performance and make comparisons among statistical and machine learning Algorithms.

Melkumova and Shatskikh in 2017, conducted a comparison of Ridge and Lasso estimators. They concluded the superiority of these methods under many circumstances [1].

Thongpeth et al in 2021, conducted a comparison among linear, penalized linear and random forest algorithms using visiting costs for patients in Thailand hospitals. Depending on both original and expanded data sets, they concluded that random forest algorithm was the best method while ridge linear regression was the worst [2].

AlHakeem ,S. et al in 2023, proposed suggested a hybrid model that consist of Long Short-Term Memory "LSTM" model and Convolution Neural Network "CNN" to predicting stock price index time series for six 6 sectors in Iraq's financial markets. The suggested method was proved its superiority against other methods through its smallest mean square error [3].

Hong, et al in 2023, suggested new hybrid Convolutional Neural Network for week-ahead Daily Peak Load Forecasting. They concluded via simulation experiments that the proposed model is better than the vector auto-regressive moving average model, support vector machine model, traditional multi-layer neural network and recurrent neural network [4].

Kadhim, M., A. and Radhi, A., M. in 2023, conducted a comparison among three machine learning algorithm which is Decision Tree, Logistic Regression, Random Forest through designing a system to classify brain stroke of disease. They concluded that Random Forest achieved high accuracy in classifying [5].

Kadhim et al. in 2023, developed a system to detect Covid-19 using a Convolutional Neural Network depending on X-Ray images taken of the chest area of infected and uninfected persons. They achieved high accuracy to distinguish efficiently different cases [6].

Hussein, S.S. et al. in 2023, addressed the importance of machine learning in healthcare systems. They demonstrated effective powered tools of machine learning in Covid-19 data analysis and forecasting [7].

Ahmed, A., S. and Salah, H., A. in 2023, designed a medical decision support system to assist workers in a hospital with diagnoses diseases. They used Decision Tree Algorithm, Naïve Bayes Algorithm and the Logistic Regression in classification doctor's task. They concluded that Decision Tree Algorithm attained high accuracy in comparable with the other two algorithms [8].

In this paper we will conduct performance evaluation between Linear model, Ridge, Lasso and Convolutional Neural Network depending on fat index data and simulation experiments, to achieve this goal we divide the paper into four subsections: methodology, simulation, application section that addresses the results and finally the conclusions.

2. Methodology

2.1 Linear model (LM):

Regression analysis is a simple traditional tool used for estimating functional relationships among variables; it has many application in the medicine, science, social and natural sciences [9], [10].

The linear relationship is expressed in the form of a parametric function or a statistical model connecting the response or dependent variable and one or more explanatory or independent variables. We denote the response or dependent variable by Y and the set of explanatory variables as: $X_1, X_2, \cdots, X_p$, where p denotes the number of explanatory variables. The true or the hypothesized linear relationship between response variable and independent variables can be estimated using the regression model by depending on sample data drawn from statistical population [11], [12].
The multiple linear regression model which is a straightforward extension of the simple linear model is written as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \quad \text{……….} (1)$$

Where:
Y: is continuous dependent variable.
X_i: $\mathrm{i} = 1,2, \cdots, \mathrm{p}$ are p exploratory variables.
β_0: is the model intercept.
β_i: $\mathrm{i} = 1,2, \cdots, \mathrm{p}$ are the model parameters.
ϵ: is the model error term, usually assumed to have a normal distribution with mean zero and constant variance σ^2.
The model in (1) is linear in the p parameters; it is not necessarily linear in the X variables. The unknown parameters β_i can be estimated by ordinary least squares method which attempts to find the parameter values that minimizing the sum of squares of residual as in equation (2) [13] , [14].

$$\hat{\beta} = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{n} \left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 \right\} \quad \text{………..} (2)$$

2.2. Lasso Regression (LR):

The Lasso technique that stands for Least Absolute Shrinkage and Selection Operator was firstly derived by Tibshirani in 1996 to deal with the problems of sample size less than the number of explanatory variables and it is regarded as a highly effective tool for selecting variables from a large set of variables, frequently more than the total number of observations [15].
In order to get the regression coefficients for some variables to shrink toward zero, the Lasso technique applies a constraints to the model parameters [16]. In order to minimize over-fitting and complexity of the model, shrinkage is used as the regularization process, which involves bringing the coefficients of less important variables closer to zero. It is also known as penalized regression with L_1 regularization when a penalty term proportionate to the absolute values of the regression coefficients is included [15]. The sum of the absolute weights is the

penalty term in this model, so the lasso unknown Lasso estimated coefficients $\hat{\beta}_j$ can be obtained by minimizing this objective function:

$$\sum_{i=1}^{n}\left(y_i - \left[\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right]\right)^2 \qquad \text{………… (3)}$$

subject to $\sum_{j=1}^{p}\left|\beta_j\right| \le t(\lambda)$, where:

$t(\lambda)$: is a tuning parameter.

In other words,

$$\hat{\beta}^{LASSO} = \underset{\beta \in R^p}{\operatorname{argmin}} \, RSS + \lambda\|\beta\|_1$$

where: $\|\beta\|_1$:is the $l_1 -$ norm of vector $(\beta_1, \dots, \beta_p)$

The $l_1 -$ norm of β is defined by:

$$\|\beta\|_1 = \sum_{j=1}^{p}|\beta_j|$$

Here λ is a tuning parameter that controls the degree of shrinkage. If the value of λ is zero the model reduced to the traditional linear regression.

When $p = 2$

$|\beta_1| + |\beta_2| \le t(\lambda)$

We get only one parameter in the model. If either value of β_1 or β_2 will be equal to zero, and if the value of λ is large, then both of them will appear in the model.

2.3 Ridge Regression (RR):

It is one of the model selection methods that used in regression analysis. RR used $l_2 -$ norm regression penalty as a constraint to shrinks estimated coefficient in a model towards zero. This method is more effective than other subset selection methods such as Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) when the linear correlation among exploratory variables was very high. However, since ridge regression does not actually set any estimated parameters to zero, it can be difficult to interpret a model [17]. The RR unknown β_j can be estimated by minimizing this objective function:

$$\sum_{i=1}^{n}\left(y_i - \left[\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right]\right)^2 \qquad \text{……. (4)}$$

subject to $\sum_{j=1}^{p}\beta_j^2 \le t(\lambda)$

where:

$t(\lambda)$: is a shrinkage parameter.

In terms of vectors, we get

$$\hat{\beta}^{Ridge} = \min_{\beta \in R^p} RSS + \lambda\|\beta\|_2^2$$

Where: $\|\beta\|_2$ is the $l_2 -$ norm of vector $(\beta_1, \dots, \beta_p)$, and is defined by:

$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p}\beta_j^2}$$

Where: $\|\beta\|_2$is the $l_2 -$ norm of vector $(\beta_1, \dots, \beta_p)$. The $l_2 -$ norm of β is defined by:

$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p}\beta_j^2}$$

2.4. Convolutional Neural Network (CNN):

Convolutional neural networks abbreviated as (CNNs, or ConvNets) are a special case of feed forward artificial neural networks (ANN) and they are worked in similar manner to it, in the sense that they both are made up of neurons with learnable weights and biases. CNN networks are of the most widely used deep learning techniques. It uses the convolutional operation which is a dot product operation between weights and inputs in at least one layer. Due to their self-feature learning capability, CNN can perform classification or regression

tasks from high dimensional data, without any prior experience with the feature. It is designed to work with grid-structured inputs, like 2- dimensional image, 3- dimensional image, text ,sound data, one – dimensional time series data, sentiment analysis, genetic data and any input data that have some structure with various relationships or spatial dependencies among adjacent values [18].
The essential difference between ANN and CNN is that CNN's architecture assumes implicitly that the inputs are similar to images, which allows us to use special setting of the architecture. In particular, convolutions have translation invariance property (i.e., filters are not depending on the location). It greatly reduces the number of parameters and improves the efficiency of the forward function, and therefore makes the CNN easier to optimize and less affected by data size. CNN recognized by their strong robustness, memory capacity, nonlinear mapping ability and strong self-learning ability. These properties would make it efficiently used for classification and approximation of nonlinear functions [19].
The CNN which based on information of spatial relationships are different from the recurrent neural networks which based on information of temporal relationships, so, CNNs eliminate the need for explicit definitions of the independent variables (inputs) that should be chosen or included in the analysis, since it optimize the input output modelling [20].
In most cases, when the training data set is sufficiently large, CNN accuracy exceeds that of other machine learning techniques [21].
Unlike conventional neural networks, CNNs layers with neurons arranged according to a few dimensions: often basic 2D case, width, height, channels and number of filters.
Convolutional neural networks can perform the same tasks as conventional feed-forward neural networks, with the exception that their layer operations are spatially arranged and have sparse connections between them.
The Convolution, pooling, and Rectified Linear Unit "ReLU" are the three types of layers that are often found in any convolutional neural network. Furthermore, the last set of layers is frequently fully connected and translates to a set of output layer in appropriate way to the application [22], [20].

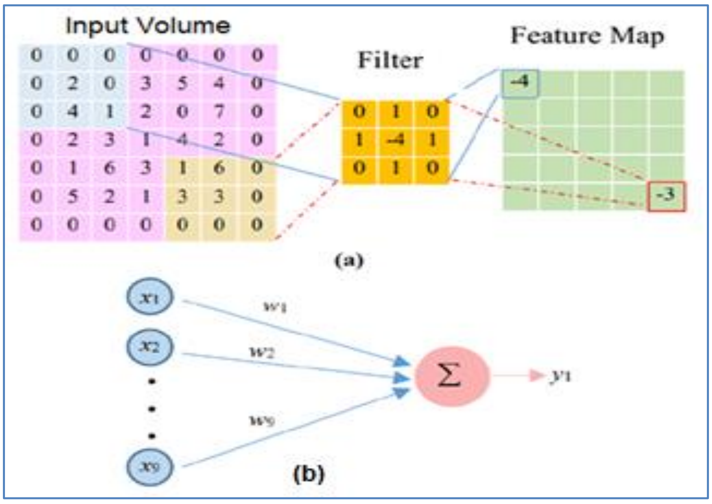


Figure 1: Convolution layer: a- input volume with a 3 × 3 filter. b- with a 3 × 3 filter, each neuron receives input from only 9 neurons of the previuos layer.

The convolutional layer extracts important exploratory variables or significant features; whereas the pooling layer reduces the number of variables which in turn further reduces the computation time. This architecture is capable of achieving self-organization. The exploratory variables are then inserted into the soft_max layer for classification or prediction purposes.

The following is a description of the prediction problem that CNN will resolve: let $X \subset R$ and $Y \subset R$ be the two random variables, X and Y have the maps as: $= f(X)$.

Given a random set of data $\{(X_i, Y_i)\}^n$, where n is the size of training data.

This prediction or approximation learning is aimed to approximate or estimate a function: $\hat{f}: X \rightarrow Y$, which can minimize the sum square of error. The squared-error loss function is defined as:

$$SSE = \sum_{i=1}^{n} \left(Y_i - \hat{f}(X_i)\right)^2 \qquad \text{……….. (5)}$$

However, then the problem is reduced to find function f that best model the relation from X to Y , described as follow:

$$f: \underset{f \subset F}{argMin} \sum_{i=1}^{n} \left(Y_i - \hat{f}(X_i)\right)^2 \qquad \text{…………. (6)}$$

CNN solves problem (6) through several steps. In the first step, X goes through a series of non-linear convolutional operations.

First, variable X is entered into the convolutional layer. Then, each successive layer x_i in the convolutional layer is calculated from the preceding layer x_{i-1} as

$$x_i = g(v_i) \qquad \text{………………. (7)}$$

$$v_i = W_i x_{i-1} + b_i \qquad \text{……….. (8)}$$

Where:

W_i: is the convolutional transform or filter.

b_i: is the bias.

g: is an activation function.

Typically, the output target value in a prediction issue is given as a real value. The activation function has many forms and, in this paper, we use the rectified linear unit (ReLU) function which is described as:

$g(v) = max(0, v)$

In the deep model, this type of nonlinear function can quickly converge in addition to producing sparse variables [23].

Each layer can eventually be expressed as the sum of the convolutions of the layer before it.

$$x_j^l = g\left(\sum_{i \in M_j} W_j^l x_i^{l-1} + b_j^l\right) \qquad \text{……… (9)}$$

The second step is using a subsampling layer through applied a pooling function. The purpose of the pooling layer is to combine variables that are closer in nature to single variable. Through this merging, the dimensions are reduced. The subsampling function is:

$$x_j^l = g\left(\beta_i^l down\left(s_i^{l-1}\right) + b_i^l\right) \qquad \text{……………………… (10)}$$

Where:

down(·): is a subsampling function

β_i^l ,b_i^l: are a multiplicative bias and an additive bias of the i^{th} point of l^{th} layer, respectively.

At the end, there will be a fully connected layer and a softmax output, as shown in Figure 2.

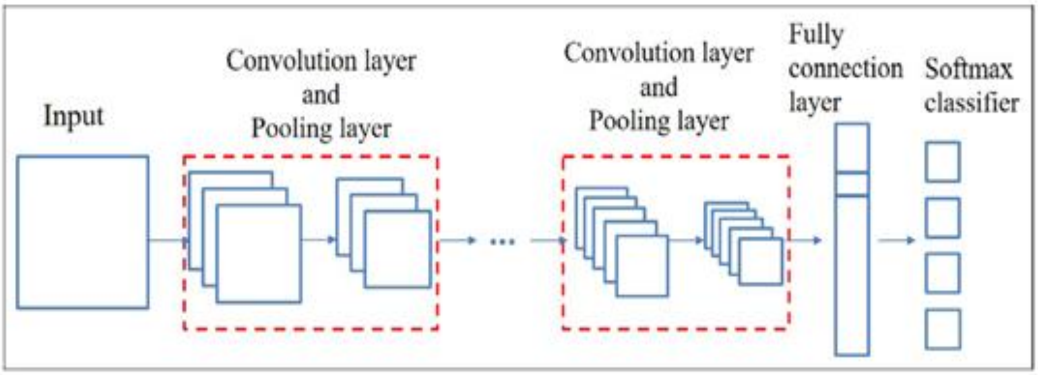


Figure 2. Architecture of a Classical Convolutional Neural Network

Since CNN's optimization problem is extremely non-convex, the Back Propagation technique "BP" is typically used to calculate gradients and to update the weights W_j [24].
Sensitivities are the errors that BP calculated from the network. It is the gradient of every unit and is defined as.:

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial v}\frac{\partial v}{\partial b} = \delta \quad \text{............ .(11)}$$

The crucial idea of BP is to propagate from upper layers to lower layer. The Equation (12) and equation (13) are shown the steps of the gradients for the l^{th} layer ($l < L$) and layer L:

$$\delta^l = (W^{l+1})^T \delta^{l+1} \cdot \dot{g}(v^l) \quad \text{............... (12)}$$

$$\delta^L = \dot{g}(v^L)(y_n - \hat{y}_n) \quad \text{.............. (13)}$$

Where: "." represents the element-by element multiplication. Subsequently, the weight can be updates by applying the δ rule, whereby the l^{th} derivatives of error about the weight are equivalent to the output product of the input vector and the sensitivity vector.

$$W^{l+1} = W^l - \eta \frac{\partial E}{\partial W^l} \quad \text{................ (14)}$$

$$\frac{\partial E}{\partial W^l} = s^{l-1}(\delta^l)' \quad \text{................ (15)}$$

Where:
s: is the estimated dependent variable from CNN.
η: is the learning rate.

3. Simulation:

In this section, a simulation experiments are conducted, a ten exploratory variable that follow normal distribution are generated. The error term generated from normal distribution with mean zero and three levels of variances $\sigma^2 = 10, 15, 20$ that represent broad range of data dispersion. The linear relationship with parameters ($\beta_0 = 1, \beta_1 = 0.5, \beta_2 = 1, \beta_3 = 3, \beta_4 = 2, \beta_5 = 1, \beta_6 = 3, \beta_7 = 0.5, \beta_8 = 1.5, \beta_9 = 4, \beta_{10} = 0.25$) is assumed. A different sample sizes that represent small ($n = 30$), moderate ($n = 50, 100$) and largr scale ($n = 200, 400$) are used to discover the behaviour of the mentioned methods. After partitioning the data into training set that consists of 80% and the rest for testing set, the Mean Square Error MSE and Mean Absolute Error MAE are computed for training and testing sets.

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \quad \text{.................... (16)}$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \quad \text{..................... (17)}$$

Table 1: MSE and MAE for Linear, Ridge, Lasso, CNN and CNN_Lasso $\sigma^2 = 10$.

sample size	Method	Training		Testing	
		MSE	**MAE**	**MSE**	**MAE**
30	Linear	0.5488	0.5833	1.644	0.9214
	Ridge	0.5567	0.5973	1.4897	0.8636
	Lasso	0.8434	0.7534	1.474	0.9704
	CNN	0.7303	0.6997	1.4571	0.9739
	CNN_Lasso	0.7345	0.7014	1.5075	0.9711
50	Linear	0.8265	0.7391	1.494	1.1161
	Ridge	0.8533	0.7409	1.2922	1.0458
	Lasso	1.0813	0.7643	0.5747	0.6715
	CNN	1.05	0.7578	0.6077	0.7002
	CNN_Lasso	1.0253	0.7469	0.5919	0.6879
100	Linear	0.775	0.6727	1.3171	0.9811
	Ridge	0.7828	0.6817	1.2887	0.9742
	Lasso	0.9542	0.7736	1.149	0.9156
	CNN	0.8923	0.7498	1.1394	0.9024
	CNN_Lasso	0.9097	0.7583	1.1596	0.9166
200	Linear	0.9167	0.7689	1.066	0.8954
	Ridge	0.9172	0.7688	1.0528	0.8898
	Lasso	1.0094	0.802	0.938	0.8073
	CNN	0.9983	0.7985	0.9331	0.8063
	CNN_Lasso	0.9945	0.7941	0.9256	0.8045
400	Linear	0.9301	0.7641	1.0487	0.8167
	Ridge	0.9303	0.7639	1.0477	0.8156
	Lasso	0.9839	0.7911	1.0581	0.8288
	CNN	0.9637	0.7804	1.068	0.8228
	CNN_Lasso	0.9652	0.7816	1.0427	0.8215

Table 2: MSE and MAE for Linear, Ridge, Lasso , CNN and CNN_Lasso $\sigma^2 = 15$.

sample size	Method	Training		Testing	
		MSE	MAE	MSE	MAE
30	Linear	0.5767	0.598	1.7277	0.9446
	Ridge	0.585	0.6123	1.5667	0.8854
	Lasso	0.8354	0.7488	1.5052	0.9771
	CNN	0.733	0.7022	1.5033	0.9825
	CNN_Lasso	0.7424	0.7084	1.5493	0.9825
50	Linear	0.8259	0.7388	1.4928	1.1157
	Ridge	0.8526	0.7406	1.2914	1.0454
	Lasso	1.0865	0.7714	0.5542	0.6631
	CNN	1.0553	0.7647	0.5877	0.6903
	CNN_Lasso	1.0292	0.7515	0.5677	0.674
100	Linear	0.7898	0.6791	1.3422	0.9904
	Ridge	0.7977	0.6882	1.3136	0.9835
	Lasso	0.9426	0.767	1.1956	0.9361

	CNN	0.8901	0.7474	1.1826	0.9195
	CNN_Lasso	0.9058	0.7545	1.1985	0.9329
200	Linear	0.9356	0.7768	1.088	0.9046
	Ridge	0.9356	0.7767	1.0745	0.899
	Lasso	0.9361	0.8014	0.9296	0.8088
	CNN	1.0114	0.8001	0.9315	0.8093
	CNN_Lasso	1.0023	0.7981	0.9354	0.811
400	Linear	0.9417	0.7688	1.0618	0.8218
	Ridge	0.9419	0.7687	1.0607	0.8207
	Lasso	0.9841	0.7881	1.0576	0.8274
	CNN	0.969	0.7805	1.064	0.8219
	CNN_Lasso	0.9728	0.7837	1.0513	0.8212

Table 3: MSE and MAE for Linear, Ridge, Lasso , CNN and CNN_Lasso $\sigma^2 = 20$.

sample size	Method	Training		Testing	
		MSE	MAE	MSE	MAE
30	Linear	0.59	0.6048	1.7675	0.9554
	Ridge	0.5985	0.6193	1.6034	0.8955
	Lasso	0.8319	0.749	1.5189	0.9794
	CNN	0.7352	0.7033	1.5229	0.9857
	CNN_Lasso	0.7467	0.7081	1.5618	0.9874
50	Linear	0.8229	0.7374	1.4874	1.1136
	Ridge	0.8494	0.7392	1.2867	1.0435
	Lasso	1.0885	0.7744	0.5463	0.6579
	CNN	1.0597	0.7671	0.5726	0.6806
	CNN_Lasso	1.0296	0.7522	0.5578	0.6666
100	Linear	0.7958	0.6817	1.3525	0.9942
	Ridge	0.8038	0.6908	1.3237	0.9872
	Lasso	0.937	0.7631	1.2185	0.9456
	CNN	0.8883	0.7455	1.2045	0.928
	CNN_Lasso	0.9034	0.7521	1.2184	0.9406
200	Linear	0.9426	0.7797	1.0961	0.908
	Ridge	0.9431	0.7796	1.0826	0.9024
	Lasso	1.0123	0.8012	0.9262	0.8092
	CNN	1.0067	0.8003	0.9302	0.8105
	CNN_Lasso	1.0031	0.7979	0.9445	0.8165
400	Linear	0.9451	0.7702	1.0656	0.8233
	Ridge	0.9453	0.7701	1.0646	0.8222
	Lasso	0.9843	0.7864	1.0569	0.8262
	CNN	0.9713	0.78	1.0599	0.8205
	CNN_Lasso	0.9744	0.7844	1.0543	0.8219

The results appeared in Table 1 to Table 3, illustrated the MSE and MAE for the Linear, Ridge, Lasso and CNN by depending on all exploratory variables and CNN_Lasso method that depends on variables that selected from Lasso method as inputs. We can see the that:

1- The CNN is getting outperformed than all other methods for testing set as sample size increased, since it have smallest MSE and MAE because it conduct feature selection as a preprocessing stage before conduct forward propagation training.
2- The Linear model has good performance for small sample size in training sets its behavior reversed in testing set, while Ridge method was better in moderate sample sizes in training set and its behaviors also worse in testing set.
3- The CNN_Lasso performance improved as sample size increased since it a mixture method between Lasso and CNN method.

4. Application:

In this section we compare the models described in this paper through real data set analyses. We will compare the models by looking into the predictive performance.

The percentage of body fat for 252 men as determined by underwater weighing and several body circumference measures is the standard data set that considered which contains thirteen physical attributes on the body fat percentage of a persons which is Age (years), Weight (lbs), Height (inches), Neck circumference (cm), Chest circumference (cm), Abdomen circumference (cm), Hip circumference (cm), Thigh circumference (cm), Knee circumference (cm), Ankle circumference (cm), Biceps (extended) circumference (cm), Forearm circumference (cm) and Wrist circumference (cm) and The response variable is a quantitative measure of percent body fat [25].

Table 4: Descriptive statistics of body fat variables.

variables	N	Maximum	Minimum	Mean	Median	Standard deviation	Skewness	Kurtosis
BodyFat	252	47.50	0	19.15	19.20	8.36	0.14	2.64
Age	252	81.00	22.00	44.88	43.00	12.60	0.28	2.56
Weight	252	363.15	118.50	178.92	176.50	29.38	1.19	8.14
Height	252	77.75	29.50	70.14	70.00	3.66	-5.35	61.34
Neck	252	51.20	31.10	37.99	38.00	2.43	0.54	5.64
Chest	252	136.20	79.30	100.82	99.65	8.43	0.67	3.94
Abdomen	252	148.10	69.40	92.55	90.95	10.78	0.83	5.18
Hip	252	147.70	85.00	99.90	99.30	7.16	1.48	10.30
Thigh	252	87.30	47.20	59.40	59.00	5.25	0.81	5.58
Knee	252	49.10	33.00	38.59	38.50	2.41	0.51	4.01
Ankle	252	33.90	19.10	23.10	22.80	1.69	2.24	14.68
Biceps	252	45.00	24.80	32.27	32.05	3.02	0.28	3.46
Forearm	252	34.90	21.00	28.66	28.70	2.02	-0.21	3.82
Wrist	252	21.40	15.80	18.22	18.30	0.93	0.27	3.36

We can see from Table 4, that data does not contain missing values, almost all variables have positive skewness except height and Forearm have negative skewness. The variable height has tail that longer than a Gaussian distribution and has high leptokurtic property than other variables.

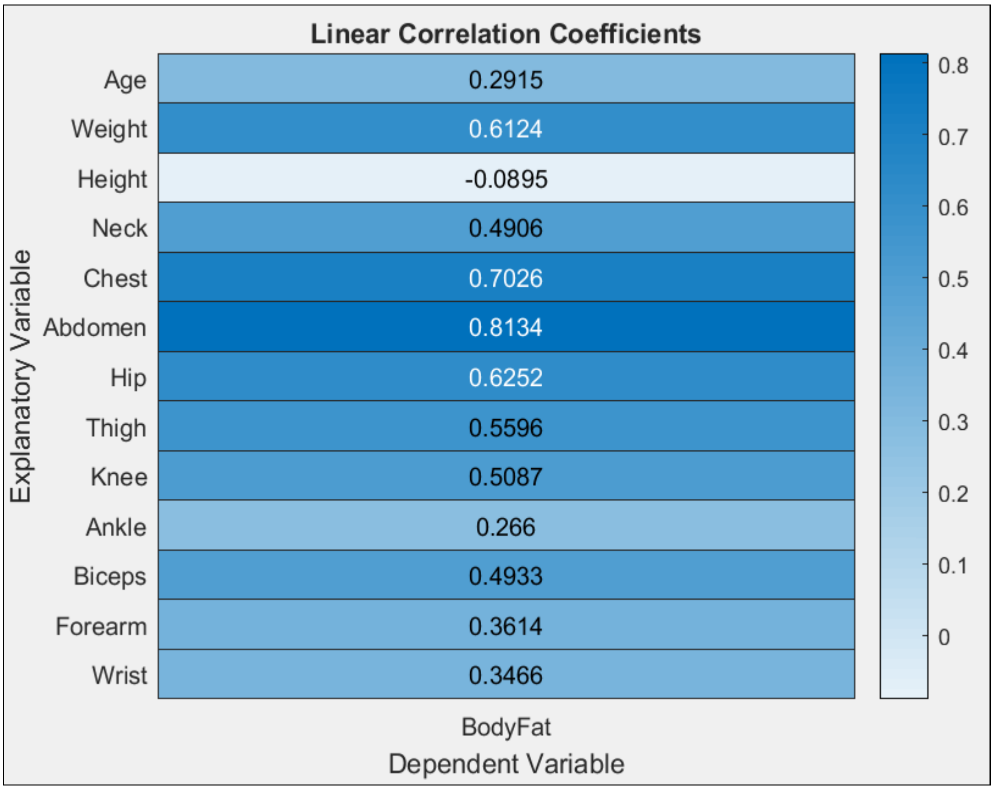


Figure 3: Linear Correlation Coefficients among variables

In Figure 3 we notice that body fat variables have high correlation with Abdomen variable than other variables. A comparison will be utilized to confirm the quality of the various models employed in this study by using the mean sum of squares errors (MSE) and mean absolute error (MAE) criteria to determine which method best fit the data against the other methods. In order to start, a training subset was randomly selected from the data set, which constitutes 80% of the whole data and the rest is the testing subset. Using the mentioned methods, the estimated models were created using the training data set. Those models were then used to predict the value of dependent variable of the testing data set and compute the MSE and MAE, the smaller the values, the better the performance.

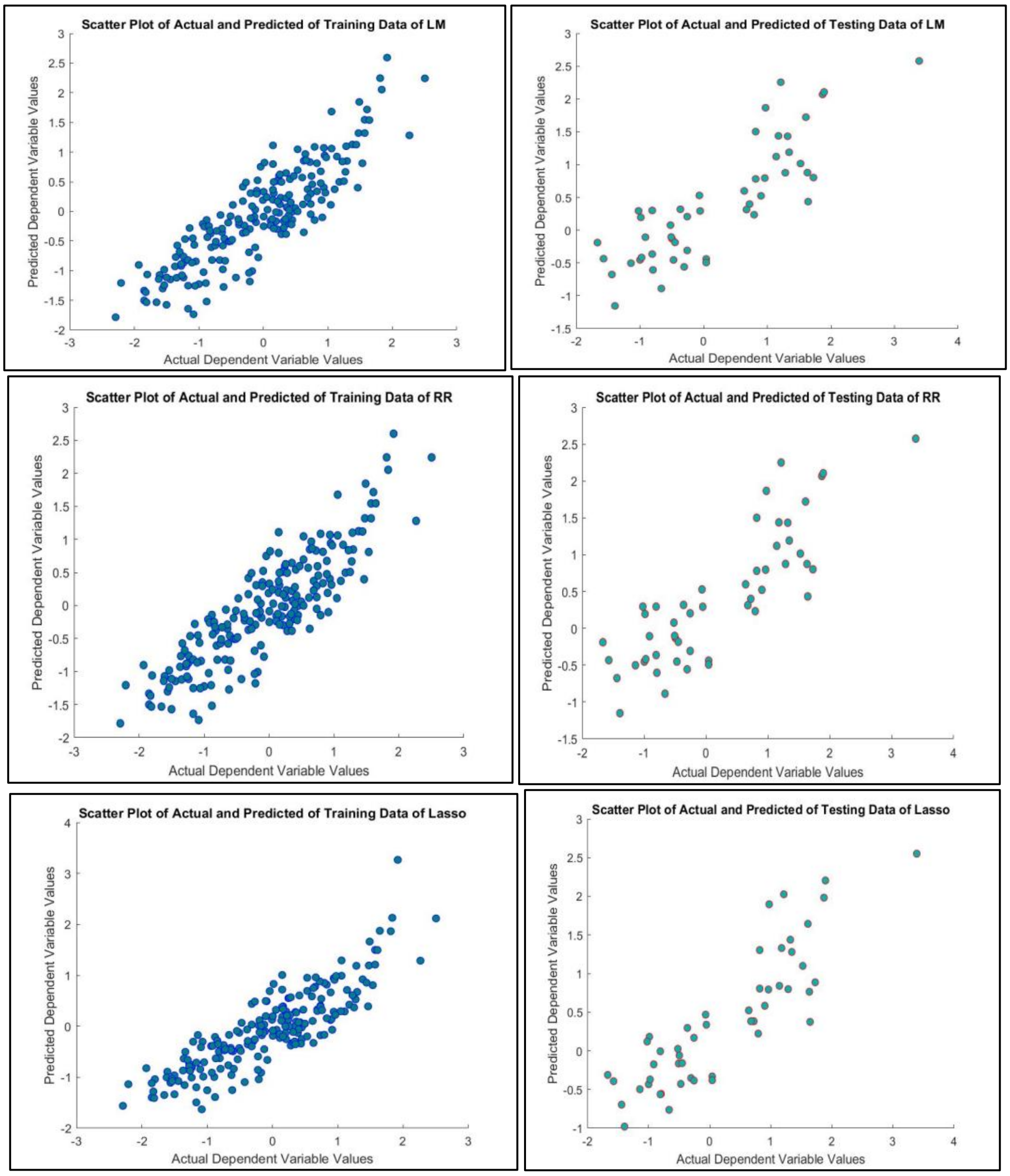
We will be using the MATLAP-R2023b software for all calculations and depending on the settings of CNN as Input Layer (13,1,1)
mini_Batch_Size= 128; Max_Epochs 60, Initial Learn Rate1e-3, Learn Rate Schedule is piecewise, Learn Rate Drop Factor 0.1, Learn Rate Drop Period 20), that allow fitting Linear, Ridge, LASSO and CNN models. To be able to validate quality of the mentioned methods, we are going to verify results on a test set and training set.

Table 5: MSE and MAE values for the training set for different techniques using all independent variables.

Method	MSE	MAE
Linear Model	0.2207	0.3928
Ridge Model	0.2207	0.3928
Lasso Model	0.2487	0.4027
CNN	0.0433	0.1714
CNN_Lasso	0.0962	0.2746

Table 6: MSE and MAE values for the testing set for different techniques using all independent variables.

Method	MSE	MAE
Linear Model	0.4025	0.5172
Ridge Model	0.4026	0.5173
Lasso Model	0.3531	0.4821
CNN	0.028	0.1297
CNN_Lasso	0.1167	0.3009



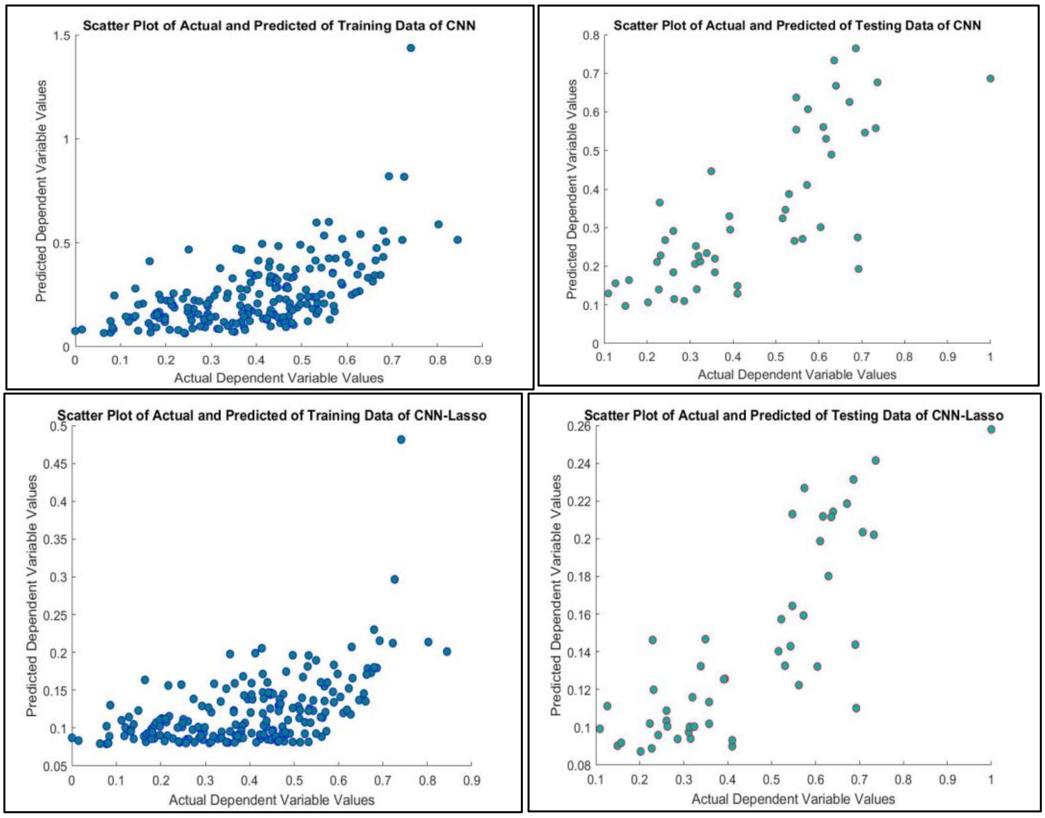


Figure 4: Scatter plot of actual and predicted values of training and testing sets for different methods

The results appeared in Table 5, Table 6 and Figure 4 illustrated the MSE and MAE for the Linear, Ridge, Lasso and CNN by depending on all exploratory variables. We can see the that:

1-CNN outperforms all other methods since it has the smallest MSE and MAE values because it conducts feature selection as a preprocessing stage before conducting forward propagation training.

2- There is a similarity in the precision of the Linear model and the Ridge model, which indicates a large correlation among exploratory variables, does not represent a serious problem and multicollinearity problem do not have large effect on the data analysis.

3-Lasso model attains significant reduction in MSE and MAE, it is be better than Linear and Ridge as a result to variable selecting that carried out during estimation. Lasso method excluded six exploratory variables which is (Weight (lbs), Chest circumference (cm), Hip circumference (cm), Thigh circumference (cm), Knee circumference (cm), Ankle circumference (cm)).

4- CNN_Lasso, which uses exploratory selection after applying the Lasso method as input variables, showed bad performance relative to CNN since it may delete important variables through the convolution process.

5-The Linear, Ridge and Lasso methods suffering from over fitting problem except CNN method, which attained good performance for testing set against training set.

5. Conclusions

The simulation Experiments for ten exploratory variables with different sample sizes and the statistical analysis of real data set that represent percentage of body fat determined by underwater weighing and various body circumference measurements leads to superiority of

CNN method over all other predictive method since it has less constraints and conduct selecting of important variables during convolution process which is important stage during the training process.

References

[1] A. Shustanov and P. Yakimov, "CNN design for real-time traffic sign recognition," *Procedia engineering,* vol. 201, pp. 718-725, 2017.

[2] W. Thongpeth, A. Lim, A. Wongpairin, T. Thongpeth, and S. Chaimontree, "Comparison of linear, penalized linear and machine learning models predicting hospital visit costs from chronic disease in Thailand," *Informatics in Medicine Unlocked,* vol. 26, p. 100769, 2021.

[3] S. AlHakeem, N. Al-Anber, and H. Atee, "Iraqi Stock Market Prediction Using Hybrid LSTM and CNN Algorithm," in *2022 Fifth College of Science International Conference of Recent Trends in Information Technology (CSCTIT)*, 2022, pp. 86-92.

[4] Y.-Y. Hong, G. F. D. Apolinario, and Y.-H. Cheng, "Week-ahead Daily Peak Load Forecasting Using Hybrid Convolutional Neural Network," *IFAC-PapersOnLine,* vol. 56, pp. 372-377, 2023.

[5] M. A. Kadhim and A. M. Radhi, "Machine Learning Prediction of Brain Stroke at an Early Stage," *Iraqi Journal of Science,* vol. 64, pp. 6596-6610, 2023.

[6] O. N. Kadhim, F. H. Najjar, and K. T. Khudhair, "Detection of COVID-19 in X-Rays by Convolutional Neural Networks," *Iraqi Journal of Science,* vol. 64, pp. 1963-1974, 2023.

[7] S. S. Hussein, W. H. Abdulsalam, and W. A. Shukur, "Covid-19 Prediction using Machine Learning Methods: An Article Review," *Wasit Journal of Pure sciences,* vol. 2, pp. 217-230, 2023.

[8] A. S. Ahmed and H. A. Salah, "A comparative study of classification techniques in data mining algorithms used for medical diagnosis based on DSS," *Bulletin of Electrical Engineering and Informatics,* vol. 12, pp. 2964-2977, 2023.

[9] R. B. Darlington and A. F. Hayes, *Regression analysis and linear models: Concepts, applications, and implementation*: Guilford Publications, 2016.

[10] N. Matloff, *Statistical regression and classification: from linear models to machine learning*: CRC Press, 2017.

[11] S. Chatterjee and A. S. Hadi, *Regression analysis by example*: John Wiley & Sons, 2013.

[12] R. A. Berk, *Statistical learning from a regression perspective* vol. 14: Springer, 2008.

[13] H. W. Altland, "Regression analysis: statistical modeling of a response variable," ed: Taylor & Francis, 1999.

[14] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction* vol. 2: Springer, 2009.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B: Statistical Methodology,* vol. 58, pp. 267-288, 1996.

[16] V. Andriopoulos and M. Kornaros, "LASSO Regression with Multiple Imputations for the Selection of Key Variables Affecting the Fatty Acid Profile of Nannochloropsis oculata," *Marine Drugs,* vol. 21, p. 483, 2023.

[17] K. L. Phillips, *An application of ridge regression and LASSO methods for model selection*: Mississippi State University, 2018.

[18] C. C. Aggarwal, "Neural networks and deep learning," *Springer,* vol. 10, p. 3, 2018.

[19] N. Y. Yen and J. C. Hung, *Frontier Computing: Theory, Technologies and Applications FC 2016* vol. 422: Springer, 2017.

[20] O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Convolutional Neural Networks," in *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, ed: Springer, 2022, pp. 533-577.

[21] X. Wang, H. Xuan, B. Evers, S. Shrestha, R. Pless, and J. Poland, "High-throughput phenotyping with deep learning gives insight into the genetic architecture of flowering time in wheat," *GigaScience,* vol. 8, p. giz120, 2019.

[22] S. Theodoridis, *Machine learning: a Bayesian and optimization perspective*: Academic press, 2015.

[23] C. Xing, L. Ma, and X. Yang, "Stacked denoise autoencoder based feature extraction and classification for hyperspectral images," *Journal of Sensors,* vol. 2016, 2016.
[24] S. Mallat, "Understanding deep convolutional networks," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences,* vol. 374, p. 20150203, 2016.
[25] A. R. Behnke and J. H. Wilmore, "Evaluation and regulation of body build and composition," Prentice-Hall*,* 1974.