Computational Economics (2023) 61:1137–1150
https://doi.org/10.1007/s10614-022-10239-5

# Prediction of Loan Rate for Mortgage Data: Deep Learning Versus Robust Regression

**Donglin Wang**[1] · **Don Hong**[1] · **Qiang Wu**[1]

Accepted: 29 January 2022 / Published online: 18 February 2022

**Abstract**
Mortgage data is often skewed, has missing information, and is contaminated by outliers. When mortgage companies or banks make prediction of note rates for new applicants, robust regression models are usually selected to deal with outliers. In this paper, we utilize deep neural network to predict the loan rate and compare its performance with three classical robust regression models. Two real mortgage data sets are used in this comparison. The results show that deep neural network has the best performance and therefore is recommended.

**Keywords** Mortgage rate prediction · Deep neural network · Huber regression · Random sample consensus · Theil–Sen regression

## 1 Introduction

The decision of mortgage loan rate is important for both applicants and mortgage companies or banks. On the one hand, a high price in mortgage loan rate could push applicants away to other companies. On the other hand, a low price could decrease the profit of the mortgage company or bank. Therefore, appropriate pricing of mortgage loan rate is critical for both lenders and borrowers. It not only provides fair loan rates to the applicants but also helps control the default risk of borrowers and thus protects the interest of mortgage companies.

✉ Donglin Wang
dwang@mtsu.edu

✉ Qiang Wu
qwu@mtsu.edu

Don Hong
dhong@mtsu.edu

[1] Department of Mathematical Sciences and Computational Science PhD Program, Middle Tennessee State University, Murfreesboro, TN, USA

Statistical analysis is an important tool to price the note rate, which is the rate the mortgage company or banks set for borrowers to calculate the principal and interest payments monthly. While the classical multiple linear regression (MLR) is often used to set up the loan rate based on the information of applicants, it has some well-known drawbacks. First, MLR is optimal only when the residuals follow a normal distribution. The prediction may be suboptimal or inaccurate when this assumption is violated which is more often for financial data sets and large data sets. Second, when the data contains highly correlated predictors, the multicollinearity among parameters can cause the unstable prediction. Third, the note rate as the response variable in the regression model is often right skewed with heavy tails which violates the assumption of normality of residuals. When MLR is used, not only outliers may inevitably be present, but the large values at the right tail may cause overestimate of note rates for most common applicants. MLR is not robust and therefore is not good at dealing with heavy tails and outliers. The prediction by multiple linear regression may be incorrect.

In order to overcome the challenges mentioned above, robust regression models are often alternatively considered to predict note rates for borrowers. Common robust models include the Huber regression model (Huber, 1992), the Random Sample Consensus (RANSAS) regression model (Fischler & Bolles, 1981), and the Theil–Sen regression model (Dang et al., 2008), to name a few. These robust models can provide more reliable prediction compared with the classical MLR, especially if there are outliers in the data sets, and therefore are able to help mortgage companies or banks make better decisions.

Deep learning (LeCun et al., 2015) as a branch of machine learning, has been shown to be successful in many fields, such as image classification (Chan et al., 2015; Perez & Wang, 2017), speech recognition (Deng et al., 2013; Noda et al., 2015), language translation (Young et al., 2018; Deng & Liu, 2018), and etc. One of the main advantages of deep learning is to use unsupervised or semi-supervised algorithms to extract features efficiently (Bengio et al., 2013). A deep feedforward neural network (Schmidhuber, 2015) consists of several layers in the model and the nodes between layers are fully connected. When it is used for regression analysis, no prior assumptions like normality of residuals are needed. This makes it a reasonable choice for the prediction of note rate for mortgage data.

Despite the drawbacks mentioned above, MLR and their variants are still the primary models used in mortgage business; see e.g. (Page, 1964; Courchane, 2007). The 2008 financial crisis has triggered increasing application of risk based mortgage pricing model (Edelberg, 2006; White, 2004; Magri & Pico, 2011; Magri, 2018). While it takes a variety of risk factors into account and adjusts for the selection bias, the prediction of mortgage rates for approved applicants is still based on linear regression models. As the great success of modern machine learning and artificial intelligence in data processing and predictive analytics is triggering significant changes in human life, business models, and industries, especially in the fields of automation, robotics, and online sales, to our best knowledge, their adoption in mortgage rate prediction seems sparse. In recent literature some machine learning approaches such as boosted regression tree, random forests, and convolutional neural networks were used for mortgage default prediction (Fitzpatrick & Mues, 2016; Kvamme et al., 2018). Although default prediction provides critical information for the mortgage companies to make decisions, they do not directly help determine the appropriate interest rates. In this paper, we apply three robust regression models and the deep feedforward neural network in mortgage data analysis to predict the note rates. Their performance is compared and the superiority of deep learning approach is illustrated by two case studies .

The rest of this paper is organized as follows. In Section 2 three classical robust models are reviewed. In Section 3 we describe the feedforward neural network. In Section 4 we describe two data sets used for this study, and perform the data analysis using the four approaches, and thoroughly compare the results. We close with the conclusions in Section 5.

## 2 Robust Regression Models

Multivariate regression analysis investigates the relationship between dependent variable $y$ and a vector of $d$ independent variables $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$. The aim is to estimate a function $\hat{f}$ given a set of $n$ paired observations $(\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d}), y_i)$, $i = 1, 2, \ldots, n$, drawn according to the model

$$y_i = f(\boldsymbol{x}_i, \boldsymbol{\theta}) + \varepsilon_i,$$

where $\boldsymbol{\theta}$ is a vector of model representation parameters and $\varepsilon_i$ are the residuals. In the mortgage rate prediction problem, the response variable is the loan rate. The predictors may include but not limited to the attributes of the applicants such as credit score and household income, the property value, the loan structure such as loan term and prepayment penalty provision, and so on. They can vary from companies to companies, from business lines to business lines, and between prime and subprime markets.

When the linear model is considered, it becomes

$$y_i = \boldsymbol{x}_i^\top \boldsymbol{\theta} + \varepsilon_i$$

and $\boldsymbol{\theta}$ is a vector of slope parameters. The classical multiple linear regression adopts least squares loss and $\boldsymbol{\theta} \in \mathbb{R}^d$ is solved by

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(y_i - \boldsymbol{x}_i{}^\top \boldsymbol{\theta}\right)^2.$$

The optimality of the classical MLR highly depends on the normality assumption of the residuals. When this assumption is violated by heavy tails or outliers, MLR may be unreliable and robust regression models stand out.

### 2.1 Huber Regression

Huber regression (Huber, 2004) uses the so-called Huber's loss to replace the least squares loss in the minimization problem to solve the parameters $\boldsymbol{\theta}$. It takes the form of

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \varphi\left(y_i - \boldsymbol{x}_i{}^T \boldsymbol{\theta}\right),$$

where $\varphi$ is the Huber's loss:

$$\varphi(t) = \begin{cases} t^2, & |t| \le \sigma \\ 2|t|\sigma - \sigma^2, & |t| > \sigma \end{cases} \tag{1}$$

and $\sigma > 0$ is a tunable parameter. Huber's loss is the same as the traditional least squares loss if the residuals are less than the threshold $\sigma$. But when the residuals are large, Huber's loss increases linearly, resulting less penalty placed on the residuals. This minimizes the impact of outliers and makes Huber regression more robust. Figure 1 illustrate the difference between Huber's loss and the least square loss.

Several other alternative losses can play a similar role and help produce robust estimators, for instance, the Tukey's biweight loss (Beaton & Tukey, 1974), the Welsch's loss (Dennis & Welsch, 1978), and the Cauchy loss (Geman & McClure, 1985), to name just a few.

## 2.2 Random Sample Consensus Regression

Random sample consensus (RANSAC) regression (Fischler & Bolles, 1981) is another robust regression model. Its main idea is to split the data into two sets: the inlier set and the outlier set. A good model should be fit only with observations from the inlier set. It utilizes an iteration process to search for good models. In each iteration, a subset $S$ of $m$ of randomly selected observations is used to fit the model. On the one hand, $m$ should be minimal to reduce the possibility of including outliers and on the other hand, it should also be large enough to reliably estimate all

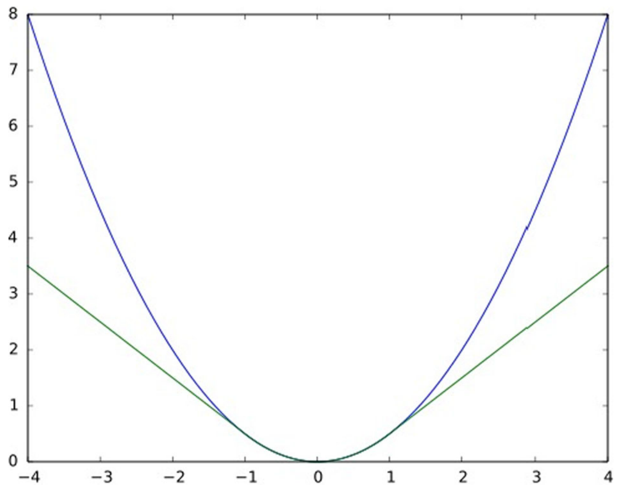


**Fig. 1** Huber loss with $\sigma = 1$ (the bottom green curve) versus the least square square loss (the upper blue curve). Figure source: https://en.wikipedia.org/wiki/Huber_loss

unknown parameters. For instance, if a multiple linear regression model is fit, $m$ should be at least larger than the dimension of the data. Next, residuals are calculated for all observations and a predefined tolerance level is used to split the data into inlier set and outlier set. The inlier set consisting of observations with residuals smaller than the tolerance level is called the consensus set. A new model is then refit on the consensus set. This process is repeated multiple times until a stopping criterion is met.

Depending on the accuracy needs, RANSAC can exit the iteration process and output the final model in two different ways: one can exit once a large enough consensus set is found, say, if the size of the consensus set is larger than a certain proportion of whole data set (Fischler & Bolles, 1981), or one can also iterate the model generating process multiple times, generate multiple models, and select the one with the smallest error (Forsyth & Ponce, 2012).

If one believes the proportion of ourliers is no more than $\gamma$, the probability for RANSAC algorithm to output a successful model after $T$ iterations is

$$g = (1 - (1 - \gamma)^m)^T.$$

This gives the choice of number of iterations as

$$T = \frac{\log g}{\log(1 - (1 - \gamma)^m)}.$$

The error tolerance or the threshold on the consensus set size can be selected in a numerical way or based on a prior assumption (Fischler & Bolles, 1981; Raguram et al., 2012).

The algorithm is illustrated in Algorithm 1 below.

**Algorithm 1** RANSAC regression algorithm.

```
Require:
    The number of points in subset, $m$;
    The error tolerance level, $\tau$;
    The maximum number of iterations, $T$;
    The minimal required proportion for acceptable consensus sets, $p$;
Ensure:
1: Randomly choose a subset $S$ of $m$ observations and fit a regression model based on $S$;
2: Calculate the residuals and define the consensus set $\tilde{S}$ according to the error tolerance level $\tau$;
3: Refit the classical regression model based on the consensus set $\tilde{S}$
4: Repeat Step 1 to Step 3 until $\tilde{S}$ has more than $pn$ observations or a maximum number $T$ of iterations
   have been repeated;
5: return $\hat{Y} = f(\boldsymbol{x}, \hat{\boldsymbol{\theta}})$;
```

## 2.3 Theil–Sen Regression

Theil–Sen regression (Theil, 1992; Sen, 1968; Dang et al., 2008) is a bootstrapping like approach. It estimates the model parameters many times based on randomly selected subsets and utilizes the the median of the estimated parameters to produce the final regression model. Due to its computational efficiency and outlier resistance, it has been used in a variety of scientific fields such as astronomy (Akritas et al., 1995), climatology (Romanić et al., 2015), and bitcoin price prediction (Phaladisailoed & Numnonda, 2018).

The implementation of Theil–Sen regression is illustrated in Algorithm 2. It requires to specify the number of observations $k$ in a subset, which should be at least larger than the number of features. The other parameter is the the number of subsets $H$, which controls the number of regression models.

**Algorithm 2** Theil Sen regression algorithm.

```
Require:
    The number of observations in each subset, $k$;
    The number of subsets, $H$;
Ensure:
 1: Randomly sample a subset of $k$ observations from the original data set;
 2: Fit a classical MLR model based on the subset from Step 1 to get $\hat{y} = f(X, \boldsymbol{\theta})$;
 3: Repeat Step 1 and Step 2 until $H$ models are produced.
 4: Collect $u$ estimated parameters $\boldsymbol{\theta_1} \cdots \boldsymbol{\theta_u}$;
 5: return $\hat{\boldsymbol{\theta}} = \text{median}(\boldsymbol{\theta_1} \cdots \boldsymbol{\theta_u})$;
```

## 3 Deep Neural Network (DNN)

While the history of artificial neural network learning can be traced back to 1940s, its blossom is attributed to its great success in the industry more recently. Due to the fast development of computer hardware and increasing computing power, nowadays neural network learning is able to handle large scale data and produces highly accurate models. This makes it play increasing roles in numerous scientific and business fields, especially in artificial intelligence. A typical artificial neural network includes three parts: the input layer consists of the $d$ features of the input data, with each feature represented by a node. The output layers produce values that are used for decision making. For regression analysis, it has one node and produces one value that is used to predict the response for the given input data. For classification problems, the output may contain several nodes which determine the correct class labels. The layers between the input layer and out layer are called hidden layers. A neural network can have one or multiple hidden layers. In feedforward neural networks, the nodes between layers are fully connected. Each layer is computed from its precedent layer by an affine linear mapping and an activation function. Mathematically, let $v_{l,i}$ denote the $i$-th node of the $l$-th layer and $k_l$ be the number of nodes in the $l$-the layer. With notations for the input layer denoted by $v_{0,i} = x_i$ and $k_0 = d$, for each $l \geq 1$ and $1 \leq i \leq k_l$,

$$v_{k,i} = \phi\left(\sum_{j=1}^{k_{l-1}} w_{l,i,j} v_{l-1,j} + b_{l,i}\right),$$

where $w_{l,i,j} \in \mathbb{R}$, $b_{l,i} \in \mathbb{R}$, and $\phi$ is an activation function. The most popular choices for the activation function include the tanh, sigmoid, and the rectified linear activation function (ReLU). Figure 2 shows an example of FNN with two hidden layers.

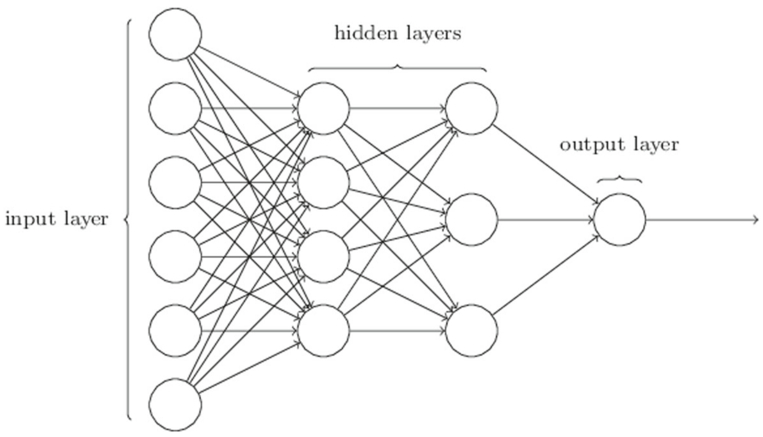


**Fig. 2** A DNN with two hidden layers, figure source: [Nielsen, 2019, Chapter 1]

The universal approximation theorem for neural networks (Hornik, 1991) asserts that all continuous functions can be approximated by shallow nets, that is, neural networks with only one hidden layer. However, there are increasing empirical evidences that neural networks with multiple layers, called deep neural networks (DNNs), are more powerful (Schmidhuber, 2015; LeCun et al., 2015; Goodfellow et al., 2016). This is also supported by some recent theoretical studies (Poggio et al., 2017; Chui et al., 2019).

The training of a neural network relies on a cost function $C(y, \hat{y})$ which measures the error when the neural network outputs a value $\hat{y}$ while the true response value (i.e. the note rate) is $y$. Typical choices for the cost function include the mean squared error function for regression analysis and cross entropy for classification. Gradient descent is usually used to implement the parameter estimation with the gradient calculated by back propagation. Let $\boldsymbol{\theta}$ denote the vector of all weights $w_{l,i,j}$ and all offset $b_{l,i}$ parameters. The update equation at the $t$-th step is

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta \frac{\partial C}{\partial \boldsymbol{\theta}},$$

where $\eta > 0$ is the learning rate (also called step size).

## 4 Case Study

Two data sets from the National Mortgage Data Repository (NMDR) are used for this study. One data set has 2180 observations and the other has 2345 observations. There are many features to describe each loan and due to similarity of features, only some of them are chosen to build the robust regression and neural network models. These independent variables are *Fair Isaac credit score*, *document of loan*, *risk type of loan*, *prepay penalty term*, *month of loan closed*, *combined loan to value ratio (CLTV)* and *length of residents*. Among these variables, *Fair Isaac credit score*, *combined loan to value ratio (CLTV)* and *length of residents* are numeric. The *document of loan*, *risk type of loan*, *prepay penalty term*, and *month of loan closed* are categorical. The *document of loan* has two types of forms (full document or not); the *risk type of loan* has four types based on the application risk grade level (AA, A+ and A-, B, C and C-); the *prepay penalty term* has three types based on the terms (none, 1 to 3 years, 5 years); and the *month of loan* refers to the time from loan application to close. Although it is is a numerical variable, it has only three possible values (3 months, 5 months, 6 months) and it is more appropriate to treat it as categorical variable. The variable *noterate* is the response variable. Table 1 summaries the variables.

Figure 3 shows the histograms of response variable *noterate* of the two data sets. Both show very right skewed distributions.

In our analysis, all numeric variables are normalized so that all values fall in the range of [0, 1]. This helps convert all variables into the common scale and avoid dominance of variables with big scale. Binary dummy variables taking values 0 or 1 are used to deal with categorical variables of more than two categories. For instance, the *risk type of loan* has four types. We use three binary variables to numerically represent them: type AA is represented as (1, 0, 0), type A+ is represented as (0, 1, 0), type A- is represented as (0, 0, 1) and type B is represented as (0, 0, 0).

After preprocessing the data, $K$-fold cross-validation is used to evaluate the performance of models. It randomly splits the data into $K$ equal groups or folds. Then $K-1$ groups are used to fit the model and the prediction error on the remaining group is used to evaluate the performance of the model. Two evaluation metrics will be used. First, to minimize the impact of the heavy tails in skewed data, we adopt the mean absolute error (MAE) as the primary evaluation metric to compare robust regression approaches. The appropriateness of this metric is obvious. A large absolute error means the predicted rate is either too high or too low. If the predicted note rate is substantially higher than expected, the applicant will be pushed away. If the predicted rate is too low, many risky applicants will be attracted due to antiselection. Second, we also look at the mean residual and its 95% confidence intervals. The underlying idea is as follows: prediction error is unavoidable regardless of the models used. A low MAE itself is insufficient to measure the risk of the mortgage business. To see this, note that a model consistently underestimating the loan rates may hurt the business by reduced profitability or possible insolvency even if it has a low MAE. Therefore, it is

**Table 1** Independent and dependent variables

| | Name | Type |
|---|---|---|
| Independent | Fair Isaac credit score | Numerical |
| | Document loan | Categorical |
| | Risk type of loan | Categorical |
| | Prepay penalty term | Categorical |
| | Month of loan closed | Categorical |
| | Combined loan to value ratio (CLTV) | Numerical |
| | Length of residents | Numerical |
| Dependent | Noterate | Numerical |

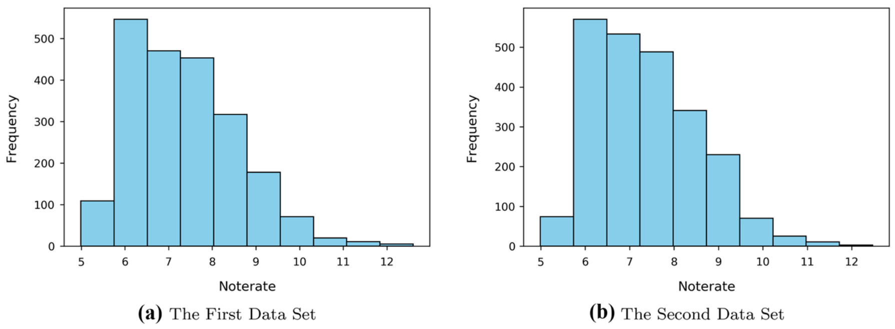


**Fig. 3** Histogram of the response variable *noterate*

necessary for the overestimate and underestimate of loan rate to be balanced and compensate each other. The mean of residuals close to 0 is a good indicator for this consideration.

In our analysis, three robust regression models are performed with the `scikit-learn` package (Pedregosa et al. 2011) in Python and all hyper-parameters are set at default values. For DNN, three hidden layers are used with 10 nodes in the first hidden layer, 8 nodes in the second hidden layer, and 6 nodes in the third hidden layer. All layers use tanh activation function. Mean square error is used as the cost function. Stochastic gradient descent algorithm with a batch size of 64 and epoch of 100 is used to train the weights.

We tried four different values of $K$ ($K = 5, 10, 15, 20$) and repeated the $K$ fold cross validation evaluation process 5 times to remove the impact of randomness for

**Table 2** MAE, mean residual, and 95% Confidence Interval for the first Data Set

| K-fold CV | Model | Mean Residual | 95% CI | MAE |
|---|---|---|---|---|
| K=5 | Huber | −0.0348 | (−0.0493, −0.0203) | 0.6110 |
| | RANSAC | −0.0973 | (−0.1133, −0.0814) | 0.6683 |
| | Theil–Sen | −0.0920 | (−0.1069, −0.0772) | 0.6242 |
| | DNN | 0.0164 | (0.0022, 0.0306) | 0.5926 |
| K=10 | Huber | −0.0348 | (−0.0492, −0.0203) | 0.6105 |
| | RANSAC | −0.1153 | (−0.1314, −0.0991) | 0.6727 |
| | Theil–Sen | −0.0939 | (−0.1088, −0.0791) | 0.6242 |
| | DNN | −0.0001 | (−0.0141, 0.0141) | 0.5863 |
| K=15 | Huber | −0.0345 | (−0.0490, −0.0201) | 0.6105 |
| | RANSAC | −0.1132 | (−0.1293, −0.0971) | 0.6664 |
| | Theil–Sen | −0.0927 | (−0.1075, −0.0778) | 0.6243 |
| | DNN | −0.0030 | (−0.0171, 0.0111) | 0.5870 |
| K=20 | Huber | −0.0346 | (−0.0491, −0.0202) | 0.6105 |
| | RANSAC | −0.1185 | (−0.1347, −0.1023) | 0.6737 |
| | Theil–Sen | −0.0928 | (−0.1076, −0.0779) | 0.6242 |
| | DNN | −0.0076 | (−0.0218, 0.0065) | 0.5873 |

**Table 3** MAE, mean residual and 95% Confidence Interval for the second Data Set

| K-fold CV | Model | Mean Residual | 95% CI | MAE |
|---|---|---|---|---|
| K=5 | Huber | −0.0304 | (−0.0445, −0.0164) | 0.6107 |
| | RANSAC | −0.1037 | (−0.1195, −0.0879) | 0.6860 |
| | Theil–Sen | −0.0969 | (−0.1114, −0.0825) | 0.6272 |
| | DNN | −0.0137 | (−0.0274, 0.0001) | 0.5932 |
| K=10 | Huber | −0.0299 | (−0.0439, −0.0159) | 0.6107 |
| | RANSAC | −0.1016 | (−0.1173, −0.0859) | 0.6807 |
| | Theil–Sen | −0.0958 | (−0.1102, −0.0814) | 0.6268 |
| | DNN | −0.0085 | (−0.0222, 0.0052) | 0.5927 |
| K=15 | Huber | −0.0302 | (−0.0442, −0.0162) | 0.6104 |
| | RANSAC | −0.0898 | (−0.1054, −0.0742) | 0.6749 |
| | Theil–Sen | −0.0960 | (−0.1104, −0.0816) | 0.6267 |
| | DNN | −0.0055 | (−0.0192, 0.0081) | 0.5865 |
| K=20 | Huber | −0.0300 | (−0.0440, −0.0160) | 0.6104 |
| | RANSAC | −0.0975 | (−0.1130, −0.0820) | 0.6719 |
| | Theil–Sen | −0.0961 | (−0.1105, −0.0817) | 0.6265 |
| | DNN | −0.0038 | (−0.0174, 0.0099) | 0.5893 |

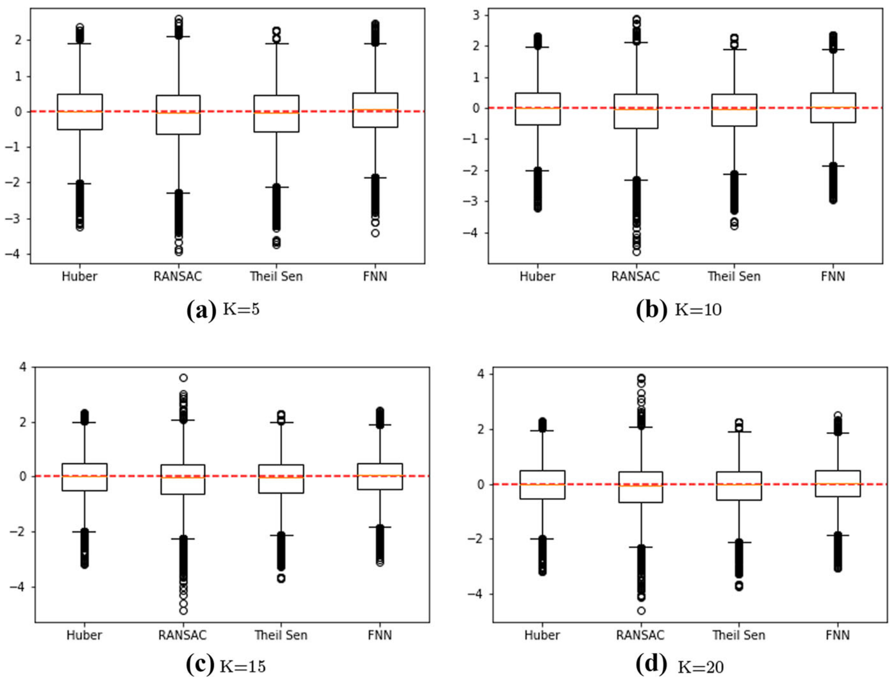
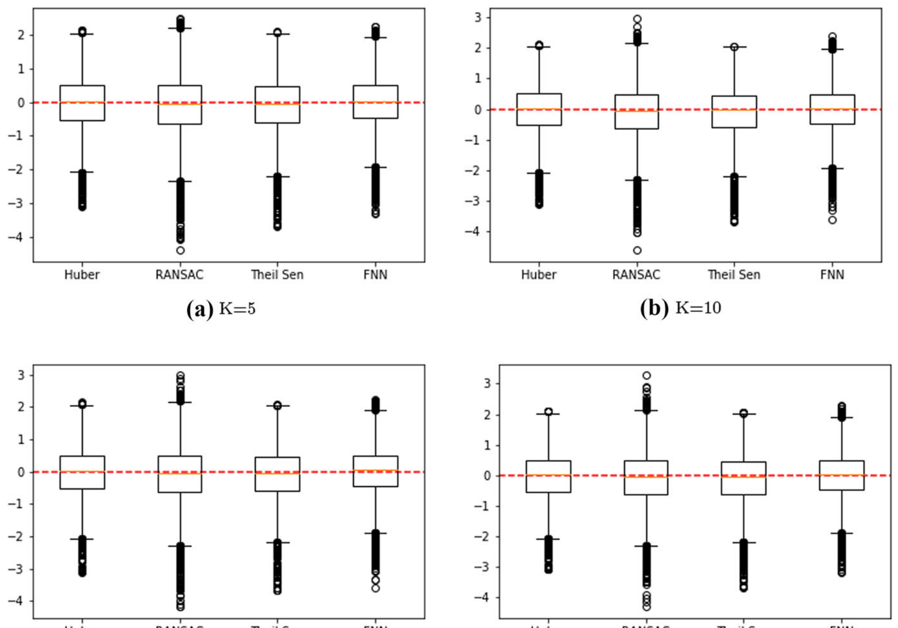


**Fig. 4** Box-plot of residuals for different K and models for 1st data



**(a)** K=5 **(b)** K=10 **(c)** K=15 **(d)** K=20

**Fig. 5** Box-plot of residuals for different K and models for 2nd data

stable comparison of the four different approaches. The MAE and mean residuals of all four approaches on the two data sets are reported in Tables 2 and 3. The results show that in all situations DNN has the smallest MAE and mean residuals closest to 0. We also performed randomization tests to compute the 95% confidence intervals for mean residual. A randomization test is robust approach to perform statistical hypothesis testing and compute confidence intervals when no prior assumption can be reasonably made on the data distribution due to the skewness; see e.g. (Givens & Hoeting, 2012). All tests are based on 5000 permutations. If the 95% confidence interval contains 0, the truth mean residual is likely to be close 0 with 95% probability. It supplements the evaluation by an empirically calculated mean residual. From the 95% confidence intervals reported in Tables 2 and 3, it seems that all three robust regression methods slightly underestimate the loan rate. In all but one case ($K = 5$ for the first data set), the 95% confidence intervals for DNN include 0, indicating that DNN predicts the loan rate at the right level.

To visualize the distribution of residuals for better understanding the performance of the four approaches, boxplots of residuals are shown in Figures 4 and 5 for the two data sets respectively. They show that Huber regression model gives the narrowest confidence interval, RANSAC model has the widest confidence interval, and DNN has the mean residual closest to 0. Moreover, DNN gives better prediction for the tails.

## 5 Conclusions

In this paper, we considered the mortgage rate prediction problem. The classical multiple linear regression is usually suboptimal due to violation of normality and presence of outliers. Three robust regression methods and deep neural networks are suggested and compared on two real data sets. Deep neural network is shown to make better prediction. It not only gives the minimal mean absolute error, but also has a mean residual closest to zero in most situations. Therefore, deep neural network is recommended for practice use by mortgage companies or banks.

**Author Contributions** All authors made equal contributions.

**Funding** This paper is partially supported by Simon Foundation (Collaboration Grant #712916) and NSF (DMS-2110826).

**Availability of data and material** Data will be available upon request.

**Code Availability** Code will be provided upon request.

## Declarations

**Conflict of interest** There is no conflict of interest.

**Ethical Approval** Not applicable.

**Consent to Participate** Not Applicable

**Consent for Publication** All authors agreed to the publication.

## References

Akritas, M. G., Murphy, S. A., & Lavalley, M. P. (1995). The theil-sen estimator with doubly censored data and applications to astronomy. *Journal of the American Statistical Association, 90*(429), 170–177.

Beaton, A. E., & Tukey, J. W. (1974). The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics, 16*(2), 147–185.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798–1828.

Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y. (2015). Pcanet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing, 24*(12), 5017–5032.

Chui, C. K., Lin, S.-B., & Zhou, D.-X. (2019). Deep neural networks for rotation-invariance approximation and learning. *Analysis and Applications, 17*(05), 737–772.

Courchane, M. (2007). The pricing of home mortgage loans to minority borrowers: How much of the apr differential can we explain? *Journal of Real Estate Research, 29*(4), 399–440.

Dang, X., Peng, H., Wang, X., & Zhang, H. (2008). Theil-sen estimators in a multiple linear regression model. Retrieved February, 2021, from http://home.olemiss.edu/~xdang/papers/MTSE.pdf.

Deng, L., Hinton, G., & Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8599–8603). IEEE.

Deng, L., & Liu, L. (2018). *Deep learning in natural language processing*. Springer.

Dennis, J. E., Jr., & Welsch, R. E. (1978). Techniques for nonlinear least squares and robust regression. *Communications in Statistics-Simulation and Computation, 7*(4), 345–359.

Edelberg, W. (2006). Risk-based pricing of interest rates in household loan markets. *Journal of Monetary Economics, 53*, 2283–2298.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*(6), 381–395.

Fitzpatrick, T., & Mues, C. (2016). An empirical comparison of classification algorithms for mortgage default prediction: evidence from a distressed mortgage market. *European Journal of Operational Research, 249*(2), 427–439.

Forsyth, D. A., & Ponce, J. (2012). *Computer vision: A modern approach*. Pearson.

Geman, S., & McClure, D. E. (1985). Bayesian image analysis: An application to single photon emmission tomography. In *Proceedings of the American Statistical Association* (pp. 12–18).

Givens, G. H., & Hoeting, J. A. (2012). *Computational statistics* (Vol. 703). Wiley.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press Cambridge. http://www.deeplearningbook.org.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks, 4*(2), 251–257.

Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics* (pp. 492–518). Springer.

Huber, P. J. (2004). *Robust statistics* (Vol. 523). Wiley.

Kvamme, H., Sellereite, N., Aas, K., & Sjursen, S. (2018). Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications, 102*, 207–217.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.

Magri, S. (2018). Are lenders using risk-based pricing in the consumer loan market? the effects of the 2008 crisis. The Effects of the 2008 Crisis (January 29, 2018). Bank of Italy Temi di Discussione (Working Paper) No, 1164.

Magri, S., & Pico, R. (2011). The rise of risk-based pricing of mortgage interest rates in italy. *Journal of Banking & Finance, 35*(5), 1277–1290.

Nielsen, M. (2019). Neural Networks and Deep Learning. http://neuralnetworksanddeeplearning.com

Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H. G., & Ogata, T. (2015). Audio-visual speech recognition using deep learning. *Applied Intelligence, 42*(4), 722–737.

Page, A. N. (1964). The variation of mortgage interest rates. *The Journal of Business, 37*(3), 280–294.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621

Phaladisailoed, T., & Numnonda, T. (2018). Machine learning models comparison for bitcoin price prediction. In *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)* (pp. 506–511). IEEE.

Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., & Liao, Q. (2017). Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing, 14*(5), 503–519.

Raguram, R., Chum, O., Pollefeys, M., Matas, J., & Frahm, J.-M. (2012). Usac: A universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 2022–2038.

Romanić, D., Ćurić, M., Jovičić, I., & Lompar, M. (2015). Long-term trends of the 'koshava'wind during the period 1949–2010. *International Journal of Climatology, 35*(2), 288–302.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks, 61*, 85–117.

Sen, P. K. (1968). Estimates of the regression coefficient based on Kendall's tau. *Journal of the American Statistical Association, 63*(324), 1379–1389.

Theil, H. (1992). A rank-invariant method of linear and polynomial regression analysis. In *Henri Theil's Contributions to Economics and Econometrics* (pp. 345–381). Springer.

White, A. M. (2004). Risk-based mortgage pricing: Present and future research. *Housing Policy Debate, 15*(3), 503–531.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine, 13*(3), 55–75.