Accepted Manuscript.
Article accepted for publication in IEEE Transactions on Engineering Management, 01/11/2019.

# A Big Data Analytics Approach for Construction Firms Failure Prediction Models

Alaka Hafiz A., *Oyedele Lukumon O., Owolabi Hakeem A., Akinade Olugbenga O., Bilal Muhammad, Ajayi Saheed O.

*Corresponding author (ayolook2001@yahoo.co.uk)

## Abstract

Using 693,000 datacells from 33,000 sample construction firms that operated or failed between 2008 and 2017, failure prediction models were developed using artificial neural network (ANN), support vector machine (SVM), multiple discriminant analysis (MDA) and logistic regression (LR). The accuracy of the models on test data surprisingly showed ANN to have only a slightly better accuracy than LR and MDA. The ANN's number of units in the hidden layer and weight decay hyperparameters were consequently tuned using the grid search. Tuning process led to tedious machine computation that was aborted after many hours without completion. The state of art Big Data Analytics (BDA) technology was, for the first time in failure prediction, consequently employed and the tuning was completed in some seconds. Mean accuracy from cross-validation was used for selection of the model with best parameter values which were used to develop a new ANN model which outperformed all previously developed models on test data. Subsequent use of selected variables to develop new models led to reduced tuning computational cost but not improved performance. Since the real-life effect of a misclassification cost is greater than the tedious computation cost, it was concluded that BDA is the best compromise.

## 1.0 Introduction

Despite holding a very high economic importance, the construction industry (was worth a staggering $7.4 trillion on the global scale in 2010 according to [1]) is well known for its high rate of business failures [2]–[4]. The reasons for this vary according to different researchers. Researchers have attributed the high failure rate to risks such as fluctuation in demand, policy changes affecting the economy, fluctuating cost of materials, high rate of litigation, safety issues, cash flow problems etc. [5]–[8]. One of the key steps taken to stem the tide of the massive failure is the development of construction firms' failure prediction models (FPM) using various tools like machine learning algorithms. The use of FPMs to identify potential construction firm failure can aid avoidance of failures as well as ensure contracts and credit, by clients and financiers respectively, are given only to healthy construction companies.

Construction firms FPM performance largely depend on the type and size of data available, and the tool used to build the FPM among other factors. To improve the all-round performance of construction firms FPMs, a large dataset and a well-tuned algorithm might be needed [9]–[11]. Data, in this case, is chiefly in the form of financial ratios which can be gotten from periodic financial statements of construction firms. Algorithm tuning has to do with altering the parameters of an algorithm until the best

model is achieved. For example, parameters like the number of hidden layers and decay in artificial neural network (ANN) or number of trees in random forest (RF), or number of iterations can be altered to improve the performance of a construction firm FPM. Tuning can however come with tedious computation cost, leading to unbearably long processing time.

Most construction firms FPM studies (e.g. [6], [12]–[16] among others) use a small number of construction firms' data, normally below 100, to build their models. A few [17]–[19] have used much higher number of construction firms' data. However, the tools in these studies were either not tuned, or the computation time was not reported.

Some general FPM studies on the other hand have attempted algorithm tuning and reported the associated computation cost which comes in form of long durations. Odom & Sharda [20] indicated that it took 24 hours to build their tuned ANN model using 191,400 iterations. Other researchers like [21] and [22] among others, required much more iterations (over 700,000 and 300,000 iterations respectively) for their model. Of these studies, only Bell [22] used over 100 firms' data (he used 1008 firms for model training). Altman et al. [23] used the data of 1000 firms and their best result, which was achieved 'after 1000 learning cycles' of ANN, required significant 'machine hours'. Though they believed a higher number of cycles could achieve a better result, the associated tedious time cost discouraged further development. Some researchers [24], [25] placed an upper limit on number of iterations to avoid the tedious computation cost.

Perhaps it can be argued that the highlighted studies are old but there is relatively recent evidence to this argument. Du Jardin [26] used a data set of 500 firms but tuned the topology, learning rate, momentum term and weight decay parameters of ANN, leading to a higher computational intensity and a much better model. As a result, "it took roughly five days to compute all network parameters with 30 PCs running Windows, and an additional day to calculate and check the final results" [26; p.2052]. With state of the art contemporary technology, such as Big Data Analytics, such a tedious computation duration can be avoided without sacrificing the necessary parameters tuning. The objectives of this study are therefore:

- To develop a high performing construction firms FPM using a well-tuned machine learning algorithm.
- To use Big Data Analytics to reduce the unfavourable waiting time usually associated with well-tuned machine learning algorithm during FPM development.

Before proceeding, below is an explanation of what is being referred to when the combination of words machine learning algorithm or Big Data Analytics are used in this paper.

Machine learning algorithm: This refers to the algorithms used in machine learning e.g. ANN, support vector machine (SVM), random forest, among others

Big Data Analytics: This refers to the framework set up to analyse a data considered to be Big Data (see sections 2 and 3 for data that qualifies as Big Data).

The next section is a brief explanation of Big Data Analytics and '**R**', which is the software used to develop the models in this study. Section 3 explains why the data used and analytics performed in this study qualify as 'Big Data'. Section 4 presents the methodology in terms of the system, data type, data source, sampling method, variables, algorithms, packages and model evaluation criteria used. Section 5 explains the initial model development attempts and how the tedious tuning process of ANN led to the decision to use BDA. Section 6 describes how the Big Data framework was set up. It also presents the analysis and results, comparing the tuned model on BDA platform to the untuned model, and comparing models from different algorithms. Section 7 gives the conclusions on the work, limitations and direction for future research.

## 2.0 Big Data Analytics and the 'R' Software

The combo of words 'Big Data' was coined by John Mashey who first used it in his Silicon Graphics (SGI) slide titled "Big Data and the Next Wave of InfraStress" [27]. Though Big Data definition is complicated since the word 'big' is relative, the Big Data concept is clearly in relation to three major characteristics of data namely: velocity, volume and variety [28]. While volume relates to the size of data, velocity relates to the data generation speed and the need for analysis of such data, and variety has to do with the extent of variability of data [28], [29]. The most common and complete Big Data framework is Apache Hadoop which is a complete open-source Big Data framework for reliable, scalable and distributed computing [30], [31]. It supports processing of huge data distributed across a cluster/assemblage of computers using simple programming model i.e. MapReduce [32].

According to R-Foundation [33], '**R**' is a free software which operates as a programming language and environment for statistical computing and visuals. It offers numerous statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, among others) and graphical techniques, and is highly extensible. It is arguably the most powerful software/platform available for data analytics. A similar, less powerful one is 'S'. Other much simpler to use software that do not require learning a programming language include WEKA and SPSS among others. These much simpler versions have many limitations including especially limited graphical outputs. The RStudio, which integrates with '**R**' as an integrated development environment, was used in this study

## 3.0 Suitability of Construction Firms FPM Data to Big Data Analytics

A dataset can be taken to be Big Data when its velocity, volume and variety become so much that current technological tools make it hard to store and/or process it [34], [35]. Its size is such that it forces a search for new approaches away from the known and trusted ones [36]. In the past, say around the 80s, it would have been a data size that required 'tape monkeys'; presently, it is a data size that will

require clusters of computer and/or cloud running concurrently and in a parallel mode to be analysed [37]. Big Data Analytics can be defined as involving analysis of huge data in order to unmask valuable patterns/information [35].

Although size is a key feature in qualifying data as 'Big Data', the nature of analysis is as important as much. Jacobs [38], in his experiment, showed why a dataset could qualify or not qualify to be classified as Big Data. Jacobs [38] created a demographic data (religion, marital status, ethnicity etc.) of the world population in a table of circa 10 columns and over 7 billion rows which was contained in a 100 gigabyte hard disk. Simple programs written to return answers to queries like the mean age of the world population ran smoothly on a computer with low performance CPU, thus not making the data viable to be classified as Big Data. An attempt to simply load the same data, without performing any analysis, on a commonly used enterprise grade database system (PostgreSQL6) running on a super performance computer (an eight core Mac Pro workstation equipped with 20 gigabyte RAM and two terabytes of RAID 0 disk) had to be aborted after six hours of unsuccessful upload. A serious analysis of the created data on this database will obviously take days if not weeks or months hence it can be classified as Big Data in this case, based mainly on analysis.

This example is why Hadley Wickham, a popular R language developer, interestingly explained that data can be classified as Big Data once its analysis by CPU (central processing unit) takes too long [39]. According to Bracht [39]. "it's not about the size of the original data set, but about the size of the biggest object created during the analysis process. Depending on the analysis type, a relatively small data set can lead to very large objects. To give an example: the distance matrix in hierarchical cluster analysis on 10.000 records contains almost 50 Million distances."

This is similar to what happened at the preliminary stage of developing our models. We put in all 29 financial ratios as variables and after successfully developing models with multiple discriminant analysis (MDA), logistic regression (LR) and support vector machine (SVM), the ANN algorithm implemented with the 'nnet' package on R failed to converge and produced unexpected relative low accuracy (see section 5). We thus decided to tune some parameters and at some point, got an error about the system not being able to allocate the required vector size required for analysis; there were also cases of the system slowing to a crawl leading to training abortion (see section 5 for more details). This was on a high spec computer with i7 processor, 16 gigabyte RAM, one terabyte hard disc and a 64-bit operating system (see subsection 4.1 for more details).

Epic [40] in his example used LR to carry out a correlation analysis between race and health care plan on the R software. This analysis took some seconds to execute. A similar analysis on the same data with Epitools, rather than LR, led to an out of workspace error, indicating that the computer is not capable of doing such analysis on the data in question. This reveals that both the type of analysis, and the type of tool can play a part in deciding what qualifies a dataset as Big Data.

# 4.0 Methodology

## *4.1 The System and Data Used*

The system used for all computations in this study was a high specification HP computer with Intel® Core™ i7-3610QM CPU that has 2.3 gigahertz base frequency (processor speed). The system features 16 gigabyte RAM, one terabyte hard disc and a 64-bit Windows 10 operating system.

The financial data of the construction firms used as sample in this study were downloaded from FAME Bureau Van Dijk financial database. The sample used contained 16,500 healthy and 16,500 failed construction firms. The target firms were those in operation or failed between 2008 and 2017. The starting year was selected to cover the year of global financial crisis while 2017 is the year in which the analysis was done. After inputting the year of operation and turnover criteria into FAME to perform a search, the 16,500 firms were selected from the search result list at random based on every other firm on the list. In essence, the first, 3rd, 5th, 7th, … firms were selected. This was done separately for failed and healthy firms to get 693,000 datacells of 33,000 construction firms sample dataset. For every selected firm (e.g. 3rd firm) with a scanty financial statement, the next firm (e.g. 4th) was used to replace it. Failed firms were simply identified as those categorised as 'dissolved' on FAME while healthy firms were those categorized as still being in operation.

## *4.2 The Variables*

As with most construction firms FPM studies, financial ratios of construction firms were used as the independent variables [41]. The 29 financial ratios provided by FAME database were used as the initial variables. These ratios are categorised as profitability, operational, structure and per employee ratios. Details of the offspring ratios of each category alongside their labels in the model are given in table 1.

**Table 1: Financial ratios category, their offspring and corresponding labels.**

| Financial ratios category | Financial ratios (variable) name | Variable identity in model |
|---|---|---|
| *Profitability ratios* | Return on Shareholders Funds (%) | R1 |
| | Return on Capital Employed (%) | R2 |
| | Return on Total Assets (%) | R3 |
| | Profit margin (%) | R4 |
| | Gross margin (%) | R5 |
| | Berry ratio | R6 |
| | EBIT margin (%) | R7 |
| | EBITDA margin (%) | R8 |
| *Operational ratios* | Net Assets Turnover | R9 |
| | Fixed Assets Turnover | R10 |
| | Interest Cover | R11 |
| | Stock Turnover | R12 |
| | Debtors Turnover | R13 |
| | Debtor Collection (days) | R14 |
| | Creditors Payment (days) | R15 |
| | Current ratio | R16 |
| | Liquidity ratio | R17 |
| | Shareholders liquidity ratio | R18 |

| Financial ratios category | Financial ratios (variable) name | Variable identity in model |
|---|---|---|
| *Structure ratios* | Solvency ratio (Asset based) (%) | R19 |
| | Solvency ratio (Liability based) (%) | R20 |
| | Asset Cover | R21 |
| | Gearing (%) | R22 |
| *Per employee ratios* | Profit per employee (unit) | R23 |
| | Turnover per employee (unit) | R24 |
| | Salaries/Turnover | R25 |
| | Average Remuneration per employee (unit) | R26 |
| | Shareholders' Funds per employee (unit) | R27 |
| | Working Capital per employee (unit) | R28 |
| | Total Assets per employee (unit) | R29 |

*EBIT: Earnings before interest and tax*

## *4.3 Algorithms and Packages*

The main tool for this study is the artificial neural network (ANN) as it was the tool that caused a tedious computation that called for the use of Big Data Analytics (see section 3 and 5). The ANN was executed with the nnet package in R. However, to allow for result comparisons, other popular tools were employed. These include MDA, LR and SVM, executed with the mda, logreg and ksvm packages in R respectively. Each package was implemented with the 'Machine Learning in R' (MLR) framework which is designed for machine learning experiments in **R.** ANN and SVM have become more popular with construction firms FPM in recent times because they seem to produce more accurate results. The data was split 70:30 for training and testing for each tool.

## *4.4 Evaluation Criteria*

The healthy and failed firms represent opposite classes in the FPMs. The word 'status' was used to represent the dependent variable. For model development, healthy firms were assigned a status value of one while failed firms were assigned zero. A code was written in the R software to generate the confusion matrix for each model's prediction. Most evaluation criteria used in this study are calculated from the generated confusion matrix. A typical confusion matrix output will present a construction firm FPM prediction result as shown in table 2.

**Table 2: A Standard confusion matrix result for a model.**

| | **Predicted class (failed firm) = 0** | **Predicted class (healthy firm) = 1** |
|---|---|---|
| **Actual class (failed firm) = 0** | True Positives (TP) | False Positives (FP) |
| **Actual class (healthy firm) = 1** | False Negatives (FN) | True Negatives (TN) |

The FPMs were evaluated based on a number of criteria as follows:

**Overall accuracy**: This is the ratio of the total number of correctly predicted classes to the total number of sample construction firms in the test data, calculated as:

$$Overall\ accuray = \frac{TN + TP}{N}$$

**Type I error:** It is the ratio of failed construction firms wrongly predicted as healthy to the total number of failed construction firms in the test data, usually expressed in percentage. Type I error is costlier than Type II error because it is better for a healthy firm to wrongly believe it is failing than vice versa. Type I error equation is:

$$\text{Type I error} = \frac{FP}{TN + FP}$$

**Type II error:** It is the ratio of healthy construction firms wrongly predicted as failed to the total number of healthy construction firms in the test data. Type II error equation is:

$$\text{Type II error} = \frac{FN}{TP + FN}$$

**Sensitivity and Specificity:** Sensitivity is the ratio of healthy construction firms correctly predicted as healthy to the total number of healthy construction firms in the test data, while Specificity is the ratio of failed construction firms correctly predicted as failed to the total number of failed construction firms. Specificity and '1- specificity' are used in plotting the receiver operator characteristic (ROC) curve. The equations for specificity and sensitivity are given below:

$$Specificity = \frac{TN}{TN + FP}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

**Receiver operator characteristic (ROC) curve:** Each model will be presented alongside its ROC curve which is a plot of sensitivity on the y-axis against specificity on the x-axis.

**Area under the curve (AUC):** This is the area under the ROC curve which is widely accepted as the best measure of the performance of a model. The AUC value of models with similar overall accuracy could even be different, making it easier to pick the better model. Since the maximum value of specificity and sensitivity which make up the axes of ROC curve are one, then the maximum AUC value, which represents excellent accuracy. A model with an AUC value below 0.5 has a less than average performance which is considered totally unacceptable.

## 5.0 Initial Model Development Attempts

Using the available 29 variables (see table 1), construction firms FPMs were built successfully with MDA, LR, SVM and ANN using the default settings of mda, logreg, ksvm and nnet packages respectively. Each of the model training took about 15 minutes except the ANN model which took about 40 minutes. The accuracy of each of the model on the test set, as presented in table 3, showed ANN to have only a slightly better accuracy than LR and MDA even though SVM and ANN are well known to perform much better than LR and MDA for construction firms FPMs (see a comprehensive review of FPM studies in [42]). This led to the tuning of nnet (ANN)

parameters, towards achieving the first objective of this study.

As explained in the main nnet package document, it is a software for feed-forward neural networks [43]. The nnet package allows flexible settings of some key ANN parameters. The ones tuned in this study include decay, number of units in the hidden layer and weight. The nnet package also allows maximum number of iterations and activation function to be dictated. The tuned parameters are defined as follows in the nnet package document [43]:

**Size:** number of units in the hidden layer.

**MaxNWts:** the maximum allowable number of weights.

**Decay:** parameter for weight decay. Default 0.

**Maxit:** maximum number of iterations.

The default number of units in the hidden layer, maximum allowable number of weights, weight decay and maximum number of iterations are 1; 1000; 0; and 100 respectively (i.e. Size = 1, MaxNWts = 1000, decay = 0 and maxit =100). The default convergence criterion is maximum number of iterations while the default activation function is softmax. This is the ANN parameter setting that produced the result in table 3.

Table 3: Accuracy of models developed with 29 variables and default parameter values.

| **Tool** | **ANN** | **SVM** | **LR** | **MDA** |
|---|---|---|---|---|
| **Overall accuracy** | 70.97% | 75.58 % | 67.3 9% | 68.67% |

## *5.1 ANN Parameter Tuning*

An inspection of the ANN model revealed that it did not achieve convergence but stopped training after the default maximum 100 iterations set by nnet. The first step taken was to increase the maximum number of iteration setting to 10,000 (maxit = 10,000) to see if convergence could be achieved and the effect of that on prediction accuracy. The subsequent training, which was completed in over an hour, achieved convergence after 1000 iterations and produced a similar result to the initial model. This was achieved with nnet's default setting of the maximum number of iterations as the convergence criterion.

The size (number of units in hidden layer) and decay parameters were subsequently tuned gradually, with size value changed at random to figures like 2, 5, 10 etc and decay to 0.5, 0.8, 2, 5 etc. The higher the figures, the more the iterations, number of weight, and time taken by the model trained. Results of prediction on test data by the trained models were however mixed, some being clearly better than the initial model (developed with default parameter values) while others not much better, and in rare cases even worse. A trial of randomly high figures of size = 80 and decay = 30 led to an error output on memory space saying: 'error: cannot allocate vector of size 340 Kb'. A decision to try some lower numbers like size = 25 and decay = 11 led to another error message saying weight of model was 37521 (i.e exceeded the default 1000 value of weight) so the MaxNWts and maxit parameters were set to 100,000 (MaxNWts = 100,000; maxit = 100,000) to avoid potential model development restrictions. These randomly high

figures, when successfully used caused model training to take hours to train. This was clearly not the best way forward as training was getting longer and parameter tuning were based mainly on guesses. With this method, to achieve the first objective of a well-tuned machine learning algorithm will be tedious and take too long, thereby defeating the second objective of reducing the unfavourable waiting time usually associated with well-tuned machine learning algorithm during construction firms FPM development.

To proceed, the auto tuning option of the MLR framework was used. Considering issues encountered with some high figures, but also that some high figures produced good accuracy, during random manual tuning, a search space for size was defined as between 1 and 50, with decay set between 0 and 20 using the 'makeNumericParam' function (note that a combination of size =35 and decay =5.4 had given one of the best prediction accuracies during random manual tuning). To avoid overfitting, the cross-validation resampling strategy was used. A 10-fold cross-validation was specified using the 'makeResampleDesc' function. A grid search optimization technique was then used for auto-tuning, implemented with the 'makeTuneControlGrid' function. MaxNWts was set 100,000 and maxit to 1,000,000 (MaxNWts = 100,000; maxit = 100,000). The activation function used was softmax. The convergence criterion was based on the maximum number of iterations. The performance measure for the selecting the best combination of tuned parameters was specified as accuracy (other measures like error rate and $R^2$ also exist). The tuning was implemented with the 'tuneParams' function and many models started getting developed, with various number of iterations and error message in some cases. The tuning continued for over 24 hours before it was aborted.

Changing from grid optimization search to the less demanding random optimization search did not solve the long duration problem. Neither did a reduction in cross validation size from 10 to 5-fold cross-validation, nor a reduction in upper limit of size and decay to 40 and 15 respectively. A change in activation function to linear or logistic also did not have an assessible impact as tuning had to be aborted after a long period.

### *5.2 Batching Attempt and Decision to use Big Data Analytics*

An attempt was made to use batching by dividing the training dataset (23100 firms data) into three (7100 firms data), each one-third representing a batch to be trained separately, but there was no such improvement in time that we could notice because we aborted the process after 12 hours. The training dataset was finally split into 10 (2,310 firms data). In this case, tuning was aborted after 5 hours, considering that the process would have to be repeated 10 times before combining the ten model. All batching attempts were done with the initially defined characteristics for tuning (i.e. size between 1 and 50, decay between 0 and 20, 10-fold cross-validation, etc. as explained in the second to the last paragraph in sub section 5.1).

To reduce the unfavourable waiting time associated with this well-tuned ANN (second objective), a big data analytics approach was used. Epic [40] explained

different ways that Big Data can be analysed on R as follows:

1) A small representative part of the data could be analysed. This could, at times, give all the information required from the data. This is more like analysing a representative sample of the chunk of data.
2) Since R loads data to memory for analysis, some cloud computing space could be rented for the computation. This will give R more space to perform computation and make it easier and faster to perform very complex analysis.
3) Data could be read into R as a table rather than as a frame as commonly done. This allows R to read in data only on demand but could lead to complications during analysis.
4) Data could be read and analysed in batches and the results combined, mimicking the map reduce framework. This is manual parallel computing which requires some advanced fundamental understanding of how R language operates.
5) The process of parallel or distributed computing using a dedicated set of packages called 'pdbR' could be used. This is a highly-advanced method used for extreme data sizes like those generated by Google, Facebook and Twiter, among other tech giants, which cannot even be stored on a single computer.

Since larger sample size increases reliability [44], option one was nullified. We also exempted option three for its potential complication. Option four was tried as explained earlier in this subsection but was not helping to achieve the second objective of this study hence it was left out. Option two was thus chosen as our data was nothing like that of tech giants like Google. This option was however implemented with a cluster of computers as against renting some cloud computing space.

## 6.0 Final Model Development

### *6.1 Setting up the Big Data Framework*

Apache Spark is the engine selected for the Big Data Analytics part of this study. An apache spark standalone cluster was setup on 21 computers in the newly built Big Data Laboratory in the University of West of England Business School. The spark binary was downloaded and installed on the system used for initial computations. On this system, Spark home was used to define a master IP with which the remaining 20 machines were added as worker nodes. To use RStudio on the Spark cluster, the sparklyr package was installed on RStudio. This created a new Spark pane which was used to connect the Spark master to the worker nodes. The sparklyr provided the dplyr backend, allowing normal R codes to be used for analysis on the Spark cluster.

## *6.2 Analysis and Results*

With the Big Data platform set up, the parameter tuning process of the ANN was run again with the same settings as given in subsection 5.1 (i.e. size limits set as 1 to 50; decay limits set as 0 to 20; 10-fold cross-validation; grid search optimization technique; softmax activation function; and maximum number of iterations as convergence criterion). It took over 100,000 iterations all together to compute all network parameters (i.e. for the many models trained during tuning). The whole process took about 40 minutes. The parameter combination returned for the best model was size = 40 and decay =7.14 with a mean accuracy of 82% from cross validation. Detailed results of all the construction firm FPMs (i.e. ANN, tuned ANN on BDA platform, SVM, MDA and LR) are presented in table 4. The ROC curves of all construction firm FPMs are also presented in figure1. The decision boundary plot for the SVM model, considering only 2 variables (R19 and R21), is presented in figure 2. The is the only type of plot offered on the MLR framework used. The two variables used were selected because their plot arguably looked more informative after comparing it to many other pairs' plot.

Table 4: Performance of the construction firms FPM developed with the 29 variables on test data.

| **Tool** | **^Tuned ANN** | **ANN *** | **SVM *** | **LR *** | **MDA *** |
|---|---|---|---|---|---|
| **Accuracy on test data (%)** | 85.14 | 70.97 | 75.58 | 67.39 | 68.67 |
| **AUC** | 0.926824 7 | 0.712374 4 | 0.85091 | 0.7366264 | 0.7344497 |
| **Type I error (%)** | 15.21 | 36.18 | 23.96 | 31.11 | 29.96 |
| **Type II error (%)** | 14.53 | 21.90 | 22.79 | 34.10 | 32.73 |

^ Tuned ANN on BDA platform

* Untuned models developed with the default parameter values of the packages used.

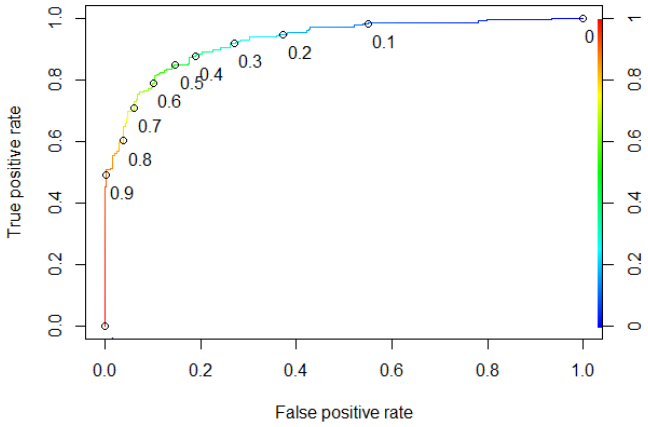


Figure 1a: ROC curve for Tuned ANN model on BDA platform using the 29 variables.

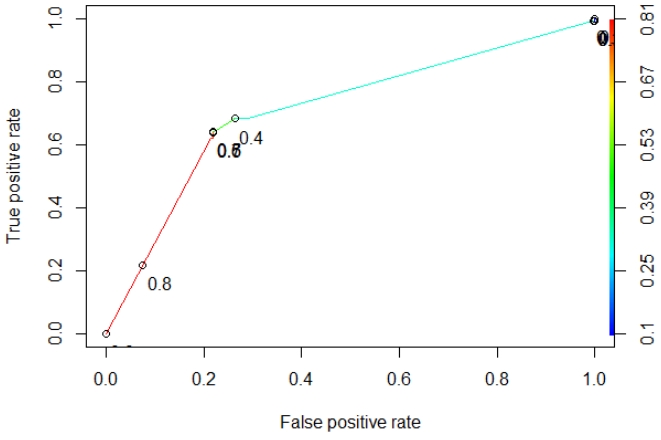


Figure 1b: ROC curve for untuned ANN model using the 29 variables.

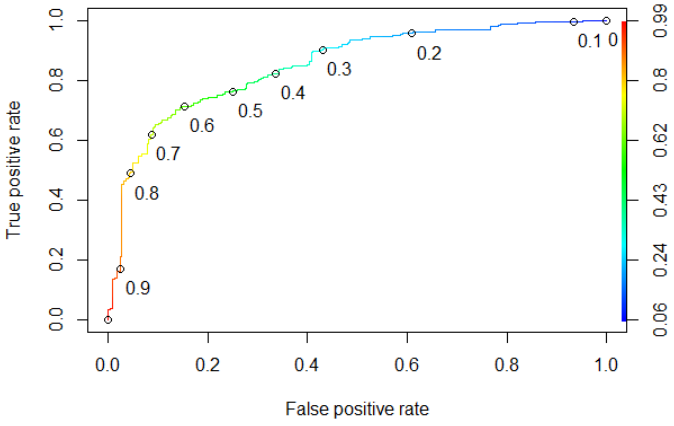


Figure 1c: ROC curve for SVM model developed using the 29 variables.

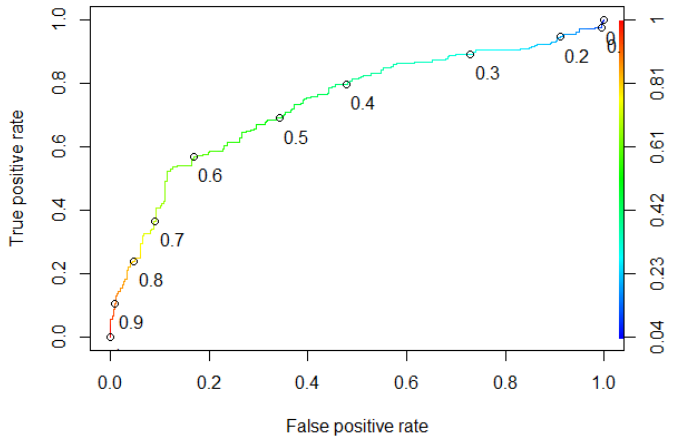


Figure 1d: ROC curve for LR model developed using the 29 variables.

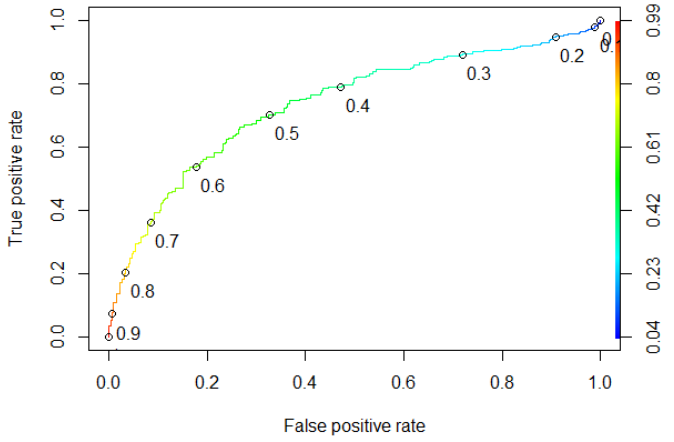


Figure 1e: ROC curve for MDA model developed using the 29 variables.

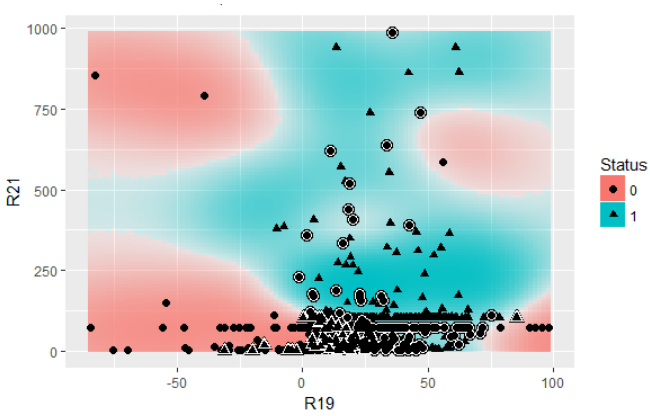


Figure 2: Decision boundary plot for the SVM model developed using the 29 variables (only 2 variables, R 29 and R21, are considered in the plot).

When dealing with a relatively large data set, the tedious process of tuning an algorithm over a reasonable search space as in this study probably shows why many studies avoid the process. The ROC curves and the corresponding AUC values however reveal the tuned ANN model to have the overall best performance accuracy (table 4 and figure 1), showing that tuning is important in the development of construction firms FPMs. Also, the accuracy of the tuned ANN model on test data bettered its mean accuracy on the 10-fold cross-validation, supporting the notion that the high AUC value (table 4) achieved by the tuned ANN means it will perform very well on new data. The difference in overall accuracy in the well-tuned and untuned model appear to significant at 15% difference, showing improved prediction on 1732 construction firms. This difference is contextually huge and its potential must be avoided, considering that the cost of using Big Data Analytics platform for proper parameter tuning of FPMs (circa $1000 even when cloud space is rented) is much lesser than the cost of wrongly financing or giving contract to a single failing construction firm, or the cost of a single construction firm's man

agement wrongly assuming the firm is healthy, there by leading to management's inactions and eventual f ailure. The associated social effects of such failure, i n terms of distress, redundancies, among others, are e ven unquantifiable [45].

Of the untuned algorithms, the SVM produced the be st construction firms FPMs. Figure 2, shows how diff icult it is to separate the 2 classes, even though it con siders only 2 variables. The symbols with white bord er in the figure indicate misclassified observations. O wing to too many observations around the same area, the misclassified observations might cover the correc tly classified observations because they are bigger du e to the white border line. The overlapping, or nearly overlapping nature of the observations within the plo t space shows the difficulty in making the right predi ctions on the data. The construction firm FPMs devel oped with LR and MDA unsurprisingly had the wors t performances but MDA surprisingly had a better Ty pe I error than Type II error (see a comprehensive re view of FPM studies in [42]).

### *6.3 Variable Selection for Potential increase in accuracy of the Construction Firms FPMs*

In an attempt to improve the accuracy of the construction firms FPMs, it was decided to use a variable selection technique to select the best predictor variables. There is no particular method that appears to have been voted as the best in literature but it is unanimously agreed that selecting the best set of variables may help to reduce multicollinearity and improve the performance of the algorithm used to develop the FPM [14], [46]–[48], among others. The random forest algorithm implemented with the cforest package on **R** was used for the variable selection process. The cforest is an implementation of the random forest and bagging ensemble algorithms utilizing conditional inference trees as base learners [49]. The default hyper parameters were used since tuning the algorithm for variable selection is out of the scope of this study. The definition and default values of the cforest algorithm parameters are given below:

1. mtry: The number of randomly preselected variables. The default is fixed to the square root of the number of input variables.

2. ntree: The number of trees (please note that default number of tree is given in the document).

The result, shown in Figure 3, displays only the top 17 variables for clarity purpose. The final seven variables, selected based on variables with a cforest value of 0.015 and above, include R11, R18, R19, R20, R21 R22 and R27.

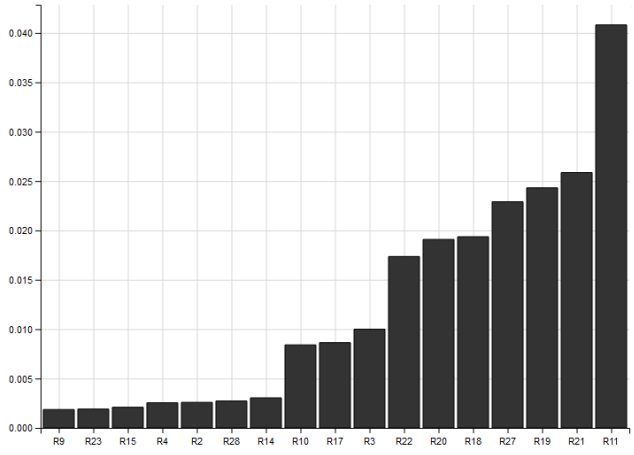


Figure 3: Variable importance according to the cforest algorithm.

The ANN was autotuned as previously described. The tuning was done twice: on the system used (see subsection 4.1) in the stand-alone form, running for around 5 hours, and on the Big Data platform, taking about 4 minutes to run. The best parameter combination returned for the best model was size = 33 and decay = 3.08 with a mean accuracy of 78.2% from cross validation. The results of the construction firms FPMs developed with the selected seven variables are presented in table 5 and figures 4a to 4e. The decision boundary plot for the SVM model, considering the 2 best variables as in figure 3 (i.e. R11 and R21), is presented in figure 5.

Table 5: Results of the construction firms FPMs developed with the seven best variables.

| **Tool** | **^Tuned ANN** | **ANN *** | **SVM *** | **LR *** | **MDA *** |
|---|---|---|---|---|---|
| **Accuracy on test data (%)** | 77.42 | 70.74 | 74.31 | 67.16 | 67.28 |
| **AUC** | 0.8576101 | 0.708132 | 0.796183 | 0.729846 | 0.728901 |
| **Type I error (%)** | 23.27 | 32.95 | 28.57 | 32.48 | 30.87 |
| **Type II error (%)** | 21.90 | 25.58 | 22.81 | 33.17 | 34.57 |

^ Tuned ANN on BDA platform

* Untuned models developed with the default parameter values of the packages used.

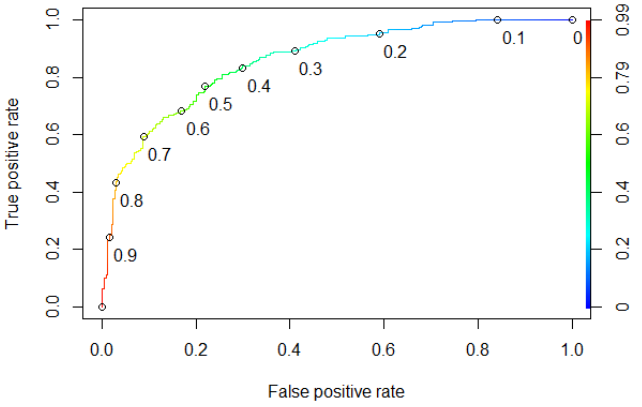


Figure 4a: ROC curve for Tuned ANN model on BDA platform developed with the 7 best variables.

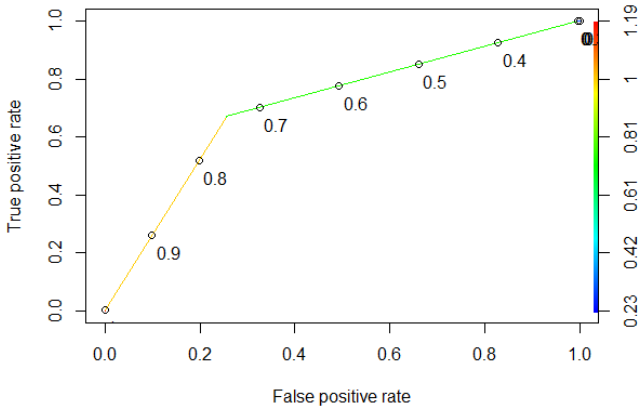


Figure 4b: ROC curve for untuned ANN model developed with the 7 best variables.

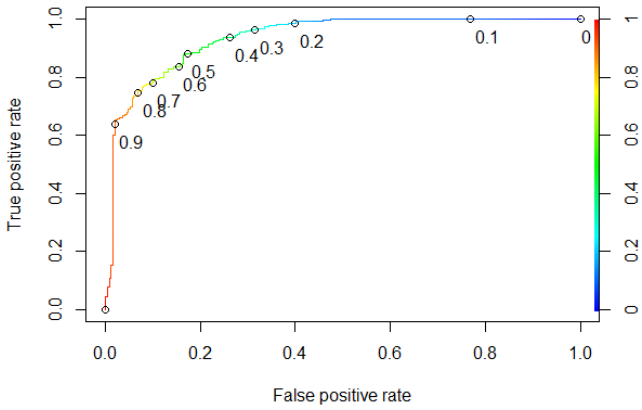


Figure 4c: ROC curve for SVM model developed with the 7 best variables.

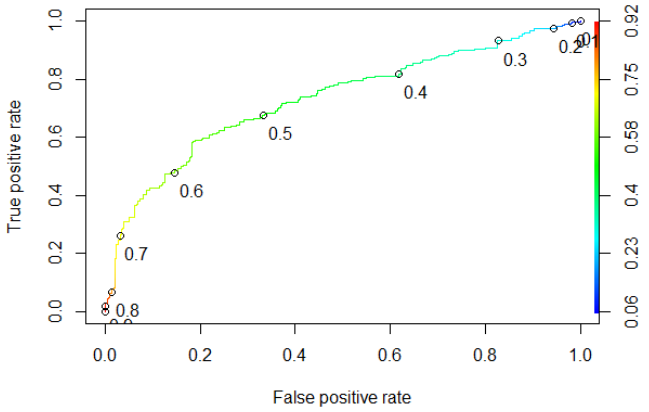


Figure 4d: ROC curve for LR model developed with the 7 best variables.

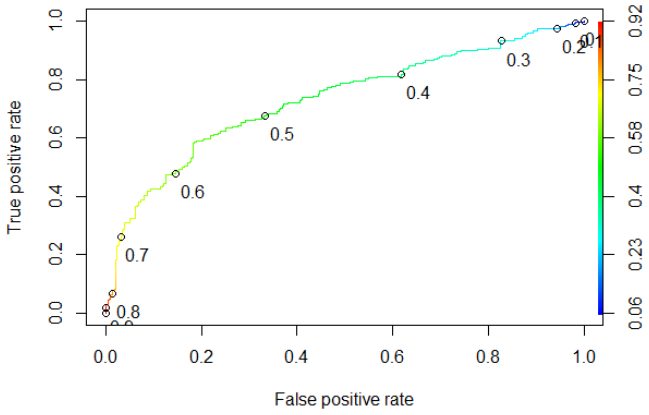


Figure 4e: ROC curve for MDA model developed wi th the 7 best variables.

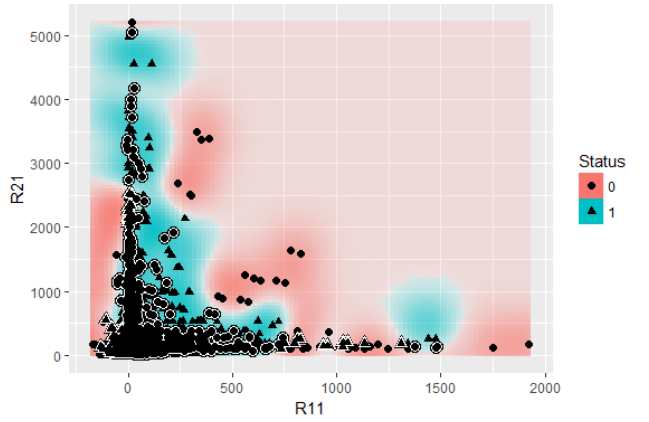


Figure 5: Decision boundary plot for the SVM model developed using the 7 best variables (only 2 variables are considered in the plot).

The reduction in time for the auto tuning of the ANN model proves that the number of variables have an impact on algorithm computation complexity. The disproportionate reduction in time between the models with 29 and 7 variables shows that either each extra variable has a multiplying effect, or depicts that certain unselected variables for the later models are difficult for the algorithm to analyse, or a combination of both cases apply. The disproportionate reduction in time between the tuned models developed on BDA platform and single system shows that the performance of the 21 computers used as BDA platform is greater than the sum of parts. The lower performance of the construction firm FPMs developed with the selected seven variables, compared to those developed with the available 29 variables, show that variable selection does not always necessarily lead to better results. This result is however highly dependent on the variable selection method used. The selection method itself can also be tuned to achieve optimal performance but this is out of scope for this study. The decision boundary plot for the SVM model still shows many observations to overlap, depicting the prediction is difficult even with the variables considered as being best by the random forest algorithm.

Although the completion of the tuning process of the ANN with 7 variables without a BDA platform in some hours makes the lesser variables number attractive, the fact that the model with 29 variables performed much better (about 7% more accurate) means it is always worth trying tuning with all the variables. As explained before, the cost of wrongly predicting just one construction firm outweighs the

cost of using the BDA platform for FPMs. Further, the overall goal is to use a much larger sample to improve reliability [44], so when a much larger sample than the one in this study is used for model development for example, the duration decrease brought about by variable selection will not be such that the model training time will be considered reasonable. Besides, although very much arguable, tuning for around 5 hours can still be considered tedious.

## 8.0 Conclusion

This paper proposed to develop construction firms FPMs using the contemporary state of the art technology BDA. It contributed to knowledge by being the first to use a well-tuned ANN algorithm to develop construction firms FPMs on BDA platform to avoid very long duration computations. The construction firms FPMs were developed using 693,000 datacells of 33,000 sample construction firms that operated or failed between 2008 and 2017.

It can be concluded that proper tuning of an algorithms can help to build a better performing construction firm FPM. A manual tuning process whereby random hyper parameter values are guessed to develop varying models for comparison is inefficient as it can take a long time train each model, especially when using large hyper parameters values on a large dataset. The auto tuning process, using an informed search space is a much better process as it allows the developer to write a set of code and leave the package to run, and develop and compare all models within the search space. The tedious computational cost, chiefly leading to excessive duration, associated such auto tuning can be reduced (i.e. the duration) by employing state of the art contemporary technology like big data analytics. The relatively high accuracy of the tuned ANN FPM in this study shows that any construction firm FMP developed without proper tuning is suboptimal and should not be adopted in practice since the financial and social cost of failure of one construction firm far outweighs the financial cost required to adopt contemporary technology that will remove any challenges of proper tuning.

Since the two key challenges to using large data are data downloading and organization as well as the extra computation cost, future studies should attempt to use Structured Query Language (SQL) to automate downloading large amount of construction firms' data and use Big Data Analytics to solve the problem of extra computation cost. The future target should be the use of hundreds of thousands of sample construction firms data with a view to develop highly reliable FPMs.

## References

[1] Global Construction Perspectives and Oxford Economics, "Global Construction 2030. A global forecast for the construction industry to 2030," Loddon, 2015.

[2] R. Kangari, "Business failure in construction industry," *J. Constr. Eng. Manag.*, vol. 114, no. 2, pp. 172–190, 1988.

[3] D. Arditi, A. Koksal, and S. Kale, "Business failures in the construction industry," *Eng. Constr. Archit. Manag.*, vol. 7, no. 2, pp. 120–132, 2000.

[4] H. Alaka, L. O. Oyedele, O. L. Toriola-Coker, H. O. Owolabi, O. O. Akinade, M. Bilal, and S. O. Ajayi, "Methodological approach of construction business failure prediction studies: a review," in *Procs 31st Annual ARCOM Conference*, 2015, pp. 1291–1300.

[5] R. J. Mason and F. C. Harris, "Predicting company failure in the construction industry," *Proc. Inst. Civ. Eng.*, vol. 66, pp. 301–7, 1979.

[6] J.-H. Chen, "Developing SFNN models to predict financial distress of construction companies," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 823–827, Jan. 2012.

[7] H. A. Alaka, L. O. Oyedele, H. A. Owolabi, S. O. Ajayi, M. Bilal, and O. O. Akinade, "Methodological approach of construction business failure prediction studies: a review," *Constr. Manag. Econ.*, vol. 34, no. 11, pp. 808–842, Nov. 2016.

[8] H. A. Alaka, L. O. Oyedele, H. A. Owolabi, M. Bilal, S. O. Ajayi, and O. O. Akinade, "Insolvency of Small Civil Engineering Firms: Critical Strategic Factors," *J. Prof. Issues Eng. Educ. Pract.*, vol. 143, no. 3, p. 4016026, Jul. 2017.

[9] S. S. Haykin, *Neural networks : a comprehensive foundation*. New Jersey: Prentice Hall, 1994.

[10] J. Min and Y. Lee, "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters," *Expert Syst. Appl.*, vol. 28, no. 4, pp. 603–614, May 2005.

[11] F.-M. Tseng and Y.-C. Hu, "Comparing four bankruptcy prediction models: Logit, quadratic interval logit, neural and fuzzy neural networks," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 1846–1853, 2010.

[12] H. L. Chen, "Model for Predicting Financial Performance of Development and Construction Corporations," *J. Constr. Eng. Manag.*, vol. 135, no. 11, pp. 1190–1200, Nov. 2009.

[13] Y. Huang, "Prediction of contractor default probability using structural models of credit risk: an empirical investigation," *Constr. Manag. Econ.*, vol. 27, no. 6, pp. 581–596, Jun. 2009.

[14] S. T. Ng, J. M. W. Wong, and J. Zhang, "Applying Z-score model to distinguish insolvent construction companies in China," *Habitat Int.*, vol. 35, no. 4, pp. 599–607, Oct. 2011.

[15] L.-K. Tsai, H.-P. Tserng, H.-H. Liao, P.-C. Chen, and W.-P. Wang, "Integration of Accounting-Based and Option-Based Models to Predict Construction Contractor Default," *J. Mar. Sci. Technol.*, vol. 20, no. 5, pp. 479–484, 2012.

[16] S. Zhanquan and G. Fox, "Large Scale Classification Based on Combination of Parallel SVM and Interpolative MDS." 2012.

[17] J. De Andrés, F. Sánchez-Lasheras, P. Lorca, and F. J. D. C. Juez, "A hybrid device of self organizing maps (SOM) and multivariate adaptive regression splines (MARS) for the forecasting of firms' bankruptcy," *Account. Manag. Inf. Syst.*, vol. 10, no. 3, pp. 351–374, 2011.

[18] F. Sánchez-Lasheras, J. de Andrés, P. Lorca, and F. J. de Cos Juez, "A hybrid device for the solution of sampling bias problems in the forecasting of firms' bankruptcy," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7512–7523, 2012.

[19] I. M. Horta and a. S. Camanho, "Company failure prediction in the construction industry," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6253–6257, Nov. 2013.

[20] M. D. Odom and R. Sharda, "A neural network model for bankruptcy prediction," in *1990 IJCNN International Joint Conference on Neural Networks*, 1990, pp. 163–168 vol.2.

[21] D. Fletcher and E. Goss, "Forecasting with neural networks," *Inf. Manag.*, vol. 24, no. 3, pp. 159–167, Mar. 1993.

[22] T. B. Bell, "Neural nets or the logit model? A

comparison of each model's ability to predict commercial bank failures," *Int. J. Intell. Syst. Accounting, Financ. Manag.*, vol. 6, no. 3, pp. 249–264, Sep. 1997.

[23] E. I. Altman, G. Marco, and F. Varetto, "Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience)," *J. Bank. Financ.*, vol. 18, no. 3, pp. 505–529, May 1994.

[24] K. A. R. Y. A. N. Tam and M. Y. Kiang, "Managerial applications of neural networks : the case of bank failure predictions *," vol. 38, no. 7, pp. 926–948, 1992.

[25] P. K. Coats and L. F. Fant, "Recognizing Financial Distress Patterns Using a Neural Network Tool," *Financ. Manag.*, vol. 22, no. 3, pp. 142–155, 1993.

[26] P. Du Jardin, "Predicting bankruptcy using neural networks and other classification methods: The influence of variable selection techniques on model accuracy," *Neurocomputing*, vol. 73, no. 10, pp. 2047–2060, 2010.

[27] F. X. Diebold, "On the Origin(s) and Development of the Term 'Big Data,'" 12–037, 2012.

[28] P. Zikopoulos and C. Eaton, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. New York: McGraw-Hill Osborne Media, 2011.

[29] M. Bilal, L. O. Oyedele, J. Qadir, K. Munir, S. O. Ajayi, O. O. Akinade, H. A. Owolabi, H. A. Alaka, and M. Pasha, "Big Data in the construction industry: A review of present status, opportunities, and future trends," *Adv. Eng. Informatics*, vol. 30, no. 3, pp. 500–521, Aug. 2016.

[30] A. Hafiz, O. Lukumon, B. Muhammad, A. Olugbenga, O. Hakeem, and A. Saheed, "Bankruptcy Prediction of Construction Businesses: Towards a Big Data Analytics Approach," in *2015 IEEE First International Conference on Big Data Computing Service and Applications*, 2015, pp. 347–352.

[31] M. Bilal, L. O. Oyedele, O. O. Akinade, S. O. Ajayi, H. A. Alaka, H. A. Owolabi, J. Qadir, M. Pasha, and S. A. Bello, "Big data architecture for construction waste analytics (CWA): A conceptual framework," *J. Build. Eng.*, vol. 6, pp. 144–156, Jun. 2016.

[32] Hadoop, "Welcome to Apache™ Hadoop®!," 2014. [Online]. Available: http://hadoop.apache.org/. [Accessed: 26-Feb-2015].

[33] R-Foundation, "R: What is R?," 2017. [Online]. Available: https://www.r-project.org/about.html. [Accessed: 20-Aug-2017].

[34] E. H. Pflugfelder and E. Helmut, "Big data, big questions," *Commun. Des. Q. Rev.*, vol. 1, no. 4, pp. 18–21, Aug. 2013.

[35] S. Suthaharan, "Big data classification," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 4, pp. 70–73, Apr. 2014.

[36] M. Bilal, L. O. Oyedele, J. Qadir, K. Munir, O. O. Akinade, S. O. Ajayi, H. A. Alaka, and H. A. Owolabi, "Analysis of critical features and evaluation of BIM software: towards a plug-in for construction waste minimization using big data," *Int. J. Sustain. Build. Technol. Urban Dev.*, vol. 6, no. 4, pp. 211–228, Oct. 2015.

[37] W. Fan and A. Bifet, "Mining big data," *ACM SIGKDD Explor. Newsl.*, vol. 14, no. 2, p. 1, Apr. 2013.

[38] A. Jacobs, "The pathologies of big data," *Commun. ACM*, vol. 52, no. 8, p. 36, Aug. 2009.

[39] O. Bracht, "Five ways to handle Big Data in R | R-bloggers." WordPress, 2013.

[40] Epic, "Big data in R," 2015. [Online]. Available: http://www.columbia.edu/~sjm2186/EPIC_R/EPIC_R_BigData.pdf. [Accessed: 20-Aug-2017].

[41] H. A. Alaka, L. O. Oyedele, H. A. Owolabi, A.

A. Oyedele, O. O. Akinade, M. Bilal, and S. O. Ajayi, "Critical factors for insolvency prediction: towards a theoretical model for the construction industry," *Int. J. Constr. Manag.*, vol. 17, no. 1, pp. 25–49, Jan. 2017.

[42] H. A. Alaka, L. O. Oyedele, H. A. Owolabi, V. Kumar, S. O. Ajayi, O. O. Akinade, and M. Bilal, "Systematic review of bankruptcy prediction models: Towards a framework for tool selection," *Expert Syst. Appl.*, vol. 94, pp. 164–184, Mar. 2018.

[43] B. Ripley and W. Venables, "Package 'nnet' Description Software for feed-forward neural networks with a single hidden layer, and for multinomial log-linear models. Title Feed-Forward Neural Networks and Multinomial Log-Linear Models." CRAN, pp. 1–11, 2016.

[44] K. S. Button, J. P. A. Ioannidis, C. Mokrysz, B. A. Nosek, J. Flint, E. S. J. Robinson, and M. R. Munafò, "Power failure: why small sample size undermines the reliability of neuroscience," *Nat. Rev. Neurosci.*, vol. 14, no. 5, pp. 365–376, May 2013.

[45] D. O'Leary, "Using neural networks to predict corporate failure," *Int. J. Intell. Syst. ...*, no. June 1997, pp. 187–197, 1998.

[46] E. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *J. Finance*, vol. 23, no. 4, pp. 589–609, 1968.

[47] A. F. Abidali and F. Harris, "A methodology for predicting company failure in the construction industry," *Constr. Manag. Econ.*, vol. 13, no. 3, pp. 189–196, May 1995.

[48] F. Edum-Fotwe, A. Price, and A. Thorpe, "A review of financial ratio tools for predicting contractor insolvency," *Constr. Manag. Econ.*, vol. 14, no. 3, pp. 189–198, May 1996.

[49] T. Hothorn, K. Hornik, C. Strobl, and A. Zeileis, "Package 'party': A Laboratory for Recursive Partytioning." CRAN, pp. 1–38, 2017.