

# Predicting Extreme Stock Performance More Accurately

Ivan Dong                      Charles Duan                      Mee-Jung Jang  
kdong@fas.harvard.edu    cduan@fas.harvard.edu    mjang@fas.harvard.edu

May 15, 2003

## Abstract

The prediction of extreme stock returns is highly useful in market analysis and trading strategies. Beneish, Lee, and Tarpley (2001) present a model using market-based signals and fundamental accounting data to make such predictions. In this paper, we use neural networks, a non-parametric predictor model, to more accurately predict extreme stock returns. We find that a well-tuned neural network can predict extreme stock returns as accurately as the probit model while requiring fewer explanatory variables, which makes for a model that is applicable to a wider range of stocks.

## 1 Introduction

The problem of predicting the financial markets is one that has captured not only the dreams of men but also the minds and journals of the academic world. In their paper “Contextual Fundamental Analysis Through the Prediction of Extreme Returns,” Beneish, Lee, and Tarpley (2001) present a novel strategy in attempting to solve this problem: they first attempt to predict *extreme performance*, that is, a stock’s performing abnormally well or poorly in comparison with the rest of the market, and then they attempt to separate winners from losers out of the pool of extreme performers.

One difficulty in this two-stage analysis is the accurate prediction of extreme stock returns. Beneish, Lee, and Tarpley propose a probit model using eighteen explanatory variables drawn from firms' recent trading characteristics and financial records. We believe that their use of a parametric model, that is, a model like probit that is based on a linear combination of its inputs, inhibits the model's predictive power, and propose using a neural network, which lacks the linear limitations of parametric models, as a more suitable alternative in predicting extreme stock performance.

In this paper, we demonstrate that a neural network can not only perform as well as the original probit model in predicting extreme stock performance but also do so while using under a third of the explanatory variables. Because of the nature of financial data, which often tends to be missing, a reduction in the number of explanatory variables used can increase the applicability of the model to more stocks.

We begin by giving a brief outline of neural networks in Section 2. In section 3 we discuss the methodology used in this study. Section 4 reproduces the probit model used in Beneish, Lee, and Tarpley and considers some of its shortcomings, Section 5 reports on the process of selecting the salient explanatory variables for the final neural network model, and Section 6 discusses the refined model itself in comparison with the probit model. We present our conclusions in Section 7.

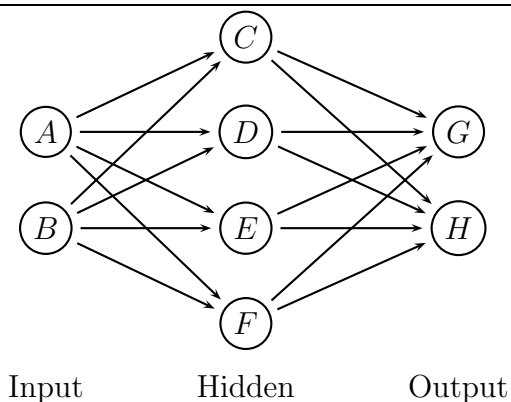
## **2 Neural Networks**

A *neural network* is a graph made up of, like any other directed graph, a set of nodes and a set of edges where each edge originates from one node

---

**Figure 1** A feed-forward neural network. Nodes  $A$  and  $B$  are inputs; nodes  $G$  and  $H$  are outputs.

---



and terminates at another. A *feed-forward neural network* is a neural network that is arranged in layers of nodes with edges only pointing “forward” through the layers, as shown in Figure 1. Neural networks are more than just graphs, however; they can be used to perform complex computational functions. In the following section we will consider how neural networks function and how they can be set up to perform statistical analysis like predicting stock performance.

Each node of the neural network serves as a mathematical function, taking input values from the incoming edges and producing output into the outgoing edges (the output then becoming input for other nodes). The node takes a weighted sum of the input values and performs a mathematical transformation on that sum, usually a logistic or step function; the result of that transformation is the output of the node on all edges. In general the same transformation function is used among all nodes; the weights on each edge that are used in calculating the weighted sum are varied to allow for the network to model different data. The problem of determining the optimum

weights for a given data set is complex and beyond the scope of this paper, but it essentially reduces to a general optimization problem.

A feed-forward neural network is measured in two dimensions: first, the number of *hidden layers*, that is, the number of layers of nodes not including the first layer where data is input or the last layer that produces the final network output; second, the *size* of each layer, or the number of nodes in a given layer. One of the most widely available and commonly used forms of neural networks is the *single-layer feed-forward network*. Such networks are generally simpler to implement, but more importantly numerous studies demonstrate that a neural network with a single hidden layer is sufficient to approximate any function given a sufficiently large size.

The ability of neural networks to approximate a wide range of functions, including non-linear ones, is important to the study of financial statistics. For example, intuitively one would expect that stocks with very high or very low prices would expect marked changes in stock price, i.e., they would be extreme performers. But a model based on linear regression, such as the probit model, is unable to provide for a situation like this one; it can only allow for either high prices or low prices to indicate extreme performance. The attractiveness of non-parametric models in general and neural networks in particular with regard to financial analysis is their ability to deal with and appropriately model such complex interactions among the data.

### **3 Overview of Methodology**

As stated earlier, the goal of our project was to improve the model for the prediction of extreme stocks presented in Beneish, Lee, and Tarpley. We did

so by first replicating their analysis and then using neural networks on the same dataset to improve the accuracy of prediction.

### 3.1 Data Collection and Organization

We used data similar to that analyzed in Beneish, Lee, and Tarpley, but were unable to construct an exact replica of their dataset. Starting from a set of company names provided by the authors of that paper, we collected data from the Compustat and CRSP databases provided by Wharton Research Data Services. We then discarded records with prices below \$5 and ADR/REIT/closed-end shares as done in their study, and additionally we discard records with missing explanatory variables.<sup>1</sup>

For each quarter, the firms in that quarter were assigned to two groups in a “1, 2, 2, 1, 1, 2, 2” pattern. This created two groups of firm-quarter records, one of which was used as an estimation data sample for analysis and the other as a holdout sample for testing that analysis.

Our dataset, after removing elements as specified above, contained 68,933 records. Beneish, Lee, and Tarpley reported 59,589 records from the same databases we used. While the discrepancy is not surprising—questions such as what constitutes missing data are not absolute and sources may have been updated since when their paper was written—it does mean that we cannot use their results directly from their report for the purpose of comparison. As a result, we reproduce the relevant parts of their analysis in Section 4, using our acquired data, to set our baseline of comparison.

Extreme performance is measured on a per-quarter basis. For each quar-

---

<sup>1</sup>The discarding of records is discussed in Section 6.2.

ter, the stocks with returns that were within the highest or lowest 2% of the quarter are designated extreme performers; the remaining 96% are designated “control” firms.<sup>2</sup> This indicator was used as our dependent variable; the independent variables were selected exactly as in the original article.

### 3.2 Method of Analysis

After establishing our baseline by replicating Beneish, Lee, and Tarpley’s probit model, we will use a neural network incorporating all of the covariates and compare its performance to the original probit model. This analysis will indicate whether the use of a neural network provides useful results in predicting extreme stock performance. In particular, one of the worries of using a neural network is overfitting the data [3]. Testing against the holdout sample will indicate whether overfitting is occurring.

In the second part of our analysis we will attempt to “clean up” the model by eliminating explanatory variables, as an excess of inputs to a neural network can actually *decrease* its predictive performance [2]. We do this by looking at first differences for each explanatory variable from the probit model and extracting only those for which there is a significant difference. We then create a new neural network using this “refined” set of explanatory variables and consider its performance in comparison to the probit model.

To generate neural networks, we use the *nnet* package in the statistical software package “R.” Table 1 gives the set of parameters we used in generating the neural networks. They are provided so that the reader may

---

<sup>2</sup>The precise details of the categorization of extreme performance, as well as those of the stock selection process, are outlined in Beneish, Lee, and Tarpley.

---

**Table 1** Parameters used in generating neural networks.

---

<b>Parameter</b>	<b>Value</b>	<b>Description</b>
rang	0.0001	Initial edge weights are uniformly distributed on $[-rang, rang]$
size	40	Number of nodes in the hidden layer of the network
skip	true	Permit formation of edges directly from inputs to outputs
entropy	true	Use entropy fitting (maximum likelihood)

---

reproduce the networks we generate in this paper; future work could consider the possibility of optimizing those parameters.

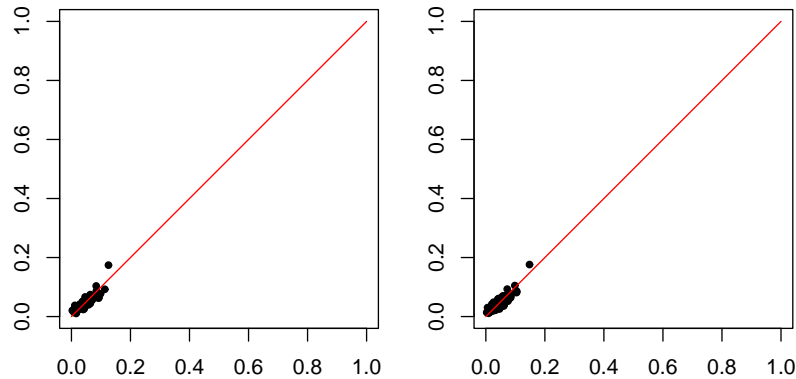
## 4 The Probit Model

Beneish, Lee, and Tarpley perform the classification of controls and extremes by probit regression on the estimate sample. There is strong evidence noted by King and Zeng that binary models such as probit suffer from bias and inefficiency if the proportion of 0s far exceed that of 1s [5]. Probit tends to underestimate the probability of extremes. Also, much of the control observations may have little relevance to the extreme cases of interest; including all the controls can degrade the predictive performance of the model by introducing random correlations. Figure 2 clearly shows these deficiencies of the probit model: for all the 34,469 observations in the estimate sample, none is predicted to have a probability of being extreme that is over 20%.

---

**Figure 2** Performance of the probit model on the estimation sample (left) and then the holdout sample (right). The data are placed into bins based on their predicted probabilities of being extreme performers. The horizontal axis represents the fraction of stocks in each bin that are extreme performers; the vertical axis is the average predicted probability of the bin.

---

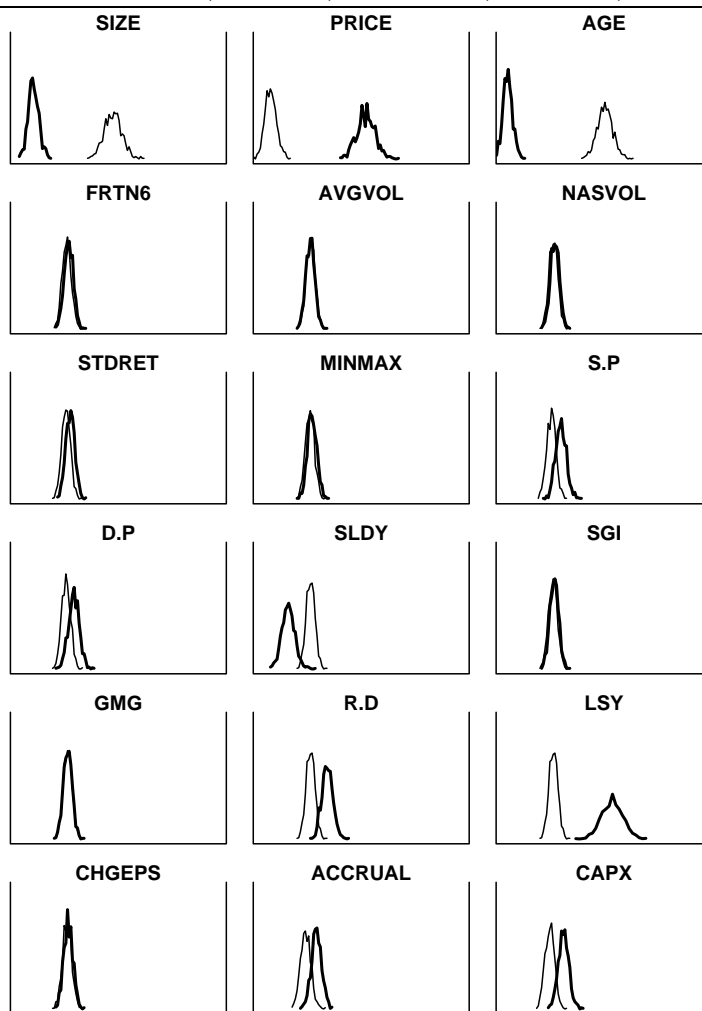


#### 4.1 The Neural Network Model

## 5 Refining the Explanatory Variable Selection

In refining our selection of explanatory variables, we took first differences based on the probabilities returned by the original probit model; these are displayed in Figure 3. In calculating these differences, we set all of the explanatory variables to their median values except for the one being tested, which was set first at its 10% quantile and then at its 90% quantile; the model was then simulated using these inputs to produce the probit probability distributions. The use of quantiles rather than preselected numerical values avoids the problem that the ranges of some variables are very small (often less than one) while the ranges of others are often in the hundreds of thousands.

**Figure 3** First differences calculated for each explanatory variable. For each graph, all variables were set to their median values except for the one listed, which was set at its 10% (thin line) and 90% (thick line) quantiles.



The graphs clearly show that the variables SIZE, PRICE, AGE, SLDY, and LSY make significant contributions to the probability of extreme performance of a stock. The other explanatory variables do not appear, from the first differences, to have such an impact on the prediction of extreme performance. Based on these observations, we choose to include these five variables only in our set of “refined” covariates.

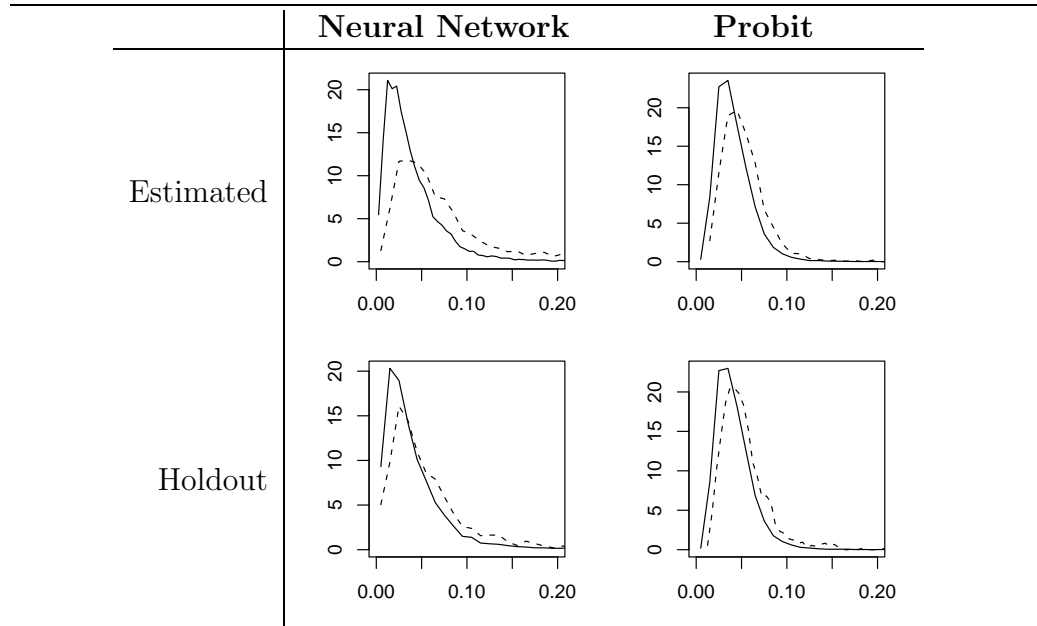
The fact that these are the most significant variables provides insights into the nature of extreme performers. Extreme performers tend to be smaller in size, higher in price, younger, and losing revenues and sales.

## **6 A Neural Network Using Refined Covariates**

We create a neural network with a hidden layer of size 40, using the five refined covariates as inputs. Figure 4 presents a visual representation of the success of this neural network at separating extreme performers from average stocks. Notice that, in fitting the dataset used to train the neural network itself, the neural network clearly outperforms the probit model in separating out the extreme performers. For the holdout sample both appear to perform at approximately the same level.

The fact that the neural network model outperforms the probit model on in-sample tests but not on out-of-sample tests indicates overfitting, which is a common problem for neural networks. However, it does help to confirm our hypothesis that the data is well-suited to a non-linear model: the neural network can fit the data more accurately with five variables than the parametric probit model can with eighteen.

**Figure 4** Comparison of the predictive powers of the neural network with reduced number of covariates. Each graph shows the distribution of probabilities for the control firm-quarters (solid) and the extreme performance firm-quarters (dashed). Greater separation indicates better prediction power.



## 6.1 A Badness Test of Model Fit

In Beneish, Lee, and Tarpley, the performance of the probit model is analyzed by using the following cost function:

$$\text{ECM} = P(E)P_I C_I + [1 - P(E)]P_{II} C_{II},$$

where  $P(E)$  is the prior probability of extreme performers (by definition 4%),  $P_I$  and  $P_{II}$  are the conditional probabilities of Type I and Type II errors, and  $C_I$  and  $C_{II}$  are the cost of those errors. A Type I error occurs when an extreme performer is classified by the model as a control stock; a Type II error occurs when a control stock is classified as an extreme performer. They choose a cutoff probability that minimizes the overall cost (ECM) given a set of Type I and Type II costs ( $C_I$  and  $C_{II}$ ).

If we can show, then, that for any value of  $P_I$  the value of  $P_{II}$  using our model is less than  $P_{II}$  for the original probit model, we prove that our model will always have a smaller overall cost (ECM) regardless of the values of  $C_I$  and  $C_{II}$ . It is thus worthwhile to consider a measure of the number of misclassified control stocks (Type II errors) for a given number of misclassified extreme stocks (Type I errors). This is equivalent to looking at a measure of misclassified control stocks for a given number of *correctly* classified extreme stocks, since the sum of the number of correctly classified and misclassified extreme stocks must be constant.

The measurement is conducted as follows. After fitting the out-of-sample data to a model, we wish to make predictions as to which of the observations are extreme performers. This is done by setting some cutoff probability  $c$

and taking all observations with a fitted probability above  $c$  to be predicted extreme performers. For a given set of fitted data  $f$ , define a function  $b_f(n)$  such that, if the cutoff is set such that  $n$  real extreme performers are correctly selected, then at least  $b_f(n)$  controls will be selected as well. The badness function for a good model should return numbers as small as possible; a perfect model should have  $b_f(n) = 0$ .

Figure 5 shows three badness functions, one for the neural net model using only refined parameters, one for a probit model including all parameters, and one for a probit model using only refined parameters. The badness was calculated against the holdout sample. This is a reasonable measure of the performance of the model, as it indicates the amount of error that will be present in decisions made by the model.

It is evident from the graph that the neural network model is essentially on par with the full-covariate probit model and significantly outperforms the probit model when it only incorporates the five refined covariates. In fact, using the neural network model over the refined-covariate probit model reduces the average badness on the range  $400 < n < 500$  by 11%, from about 6,200 incorrectly selected stocks to about 5,500. In other words, if one were to develop a trading strategy involving extreme stock returns, using the neural network model could decrease the investment cost by 11% without affecting the resulting profit significantly.

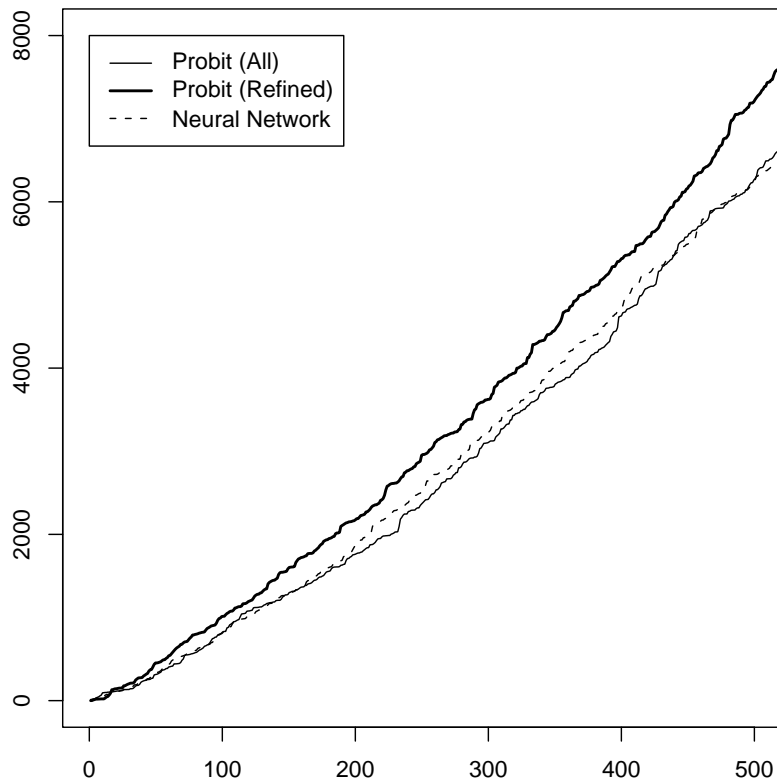
## 6.2 Why Use Fewer Explanatory Variables?

The neural network model does not, however, predict extreme stock performance more accurately than the probit model using all of the explana-

---

**Figure 5** Graphs of the badness functions for three different models against the holdout sample. The horizontal axis represents the number of extreme performers that were selected correctly, and the vertical axis indicates the number of controls that had to be selected incorrectly to achieve that number of correctly identified extreme performers.

---



tory variables. Why is it advantageous to build a model that requires fewer data?

As mentioned in Section 3.1, much of the data was discarded as a result of missing values. It is true that this could introduce bias, but we could also restrict the conclusions reached by this study to those stock-quarters that have all the necessary data available; that is, a trading strategy built out of the results of this paper would only be valid for stocks with all the data required to perform the analysis.

Another possible alternative would be to perform imputation on the missing data, thereby reducing the missing data bias. The problem with this suggestion, however, is a problem of scale. The 68,933 observations we acquired were selected out of 363,248 records in total, so over 80% of the data records retrieved from the financial databases were discarded due to their missing at least some information. As a result, imputation would be a computationally infeasible and ultimately unsuccessful task.

Part of the advantage, then, of reducing the number of relevant covariates is that more data can be imported into the analysis without the need for imputation. Additionally, if fewer data are required to make a prediction, then it is possible to make predictions on stocks that might have been missing data needed for other models. Consequently, by using a model with fewer covariates, we can both increase the accuracy of our model by adding more data to it and increase the usefulness of our model by allowing it to make predictions on more stocks.

## 7 Conclusions

The goal of this paper was to present an alternate model of predicting extreme stock performance and to demonstrate that it could make such predictions as well as the original model while requiring fewer explanatory variables. As we have shown, a neural network trained on only a properly selected set of covariates can perform just as well as a standard regression model that uses over three times the number of variables. This leads us to two conclusions: first, that much of the data being collected to predict extreme stock returns is unnecessary, and second, that the data is non-linear and best represented by a non-linear model like neural networks.

One interesting technique we used in our research methodology was the selection of significant covariates by using first differences rather than looking at coefficients. The use of first differences provides us with two advantages. First, because we take differences over the quantiles of actual data, we are able to see the difference a change *on the scale of the data* makes. A large probit coefficient for a given explanatory variable could mean either that the variable is highly significant or that the values of the variable are very small; looking at first differences highlights the former of these cases but not the latter. Second, first differences shows us distributions for given sets of explanatory variables, not just the mean values. It is possible—in fact, for the S/P and D/P variables it actually occurs, as shown in Figure 3—for the mean to shift but for the distribution to be so wide that the distributions still mostly overlap. If the distributions found for first differences for a given variable overlap, then the variable shouldn't be considered significant, even if the mean of the distributions changes. Probit coefficients cannot display

this overlap of distributions; visualizing first differences can.

Much of the research today involves adding explanatory variables to a model. While it is difficult to argue in general with the principle that more data is better, more data in this case can inhibit the model due to the missing data problem. Adding too many explanatory variables to a model obscures its usability, as it puts a burden on the user of that model to dig up all the data in order to make the model useful.

By considering more complex models such as neural networks, we hope to create a model that doesn't sacrifice predictive power but works with the benefit of requiring less data to make those predictions, thus relieving that burden of data collection. The five variables we select, firm size, stock price, firm age, and reports of sales and revenue losses, should be feasible for anyone to acquire, while variables that we discard, such as research and development expenses and capital expenditures, may be more difficult to acquire. A model is only as good as it is useful, and a model requiring too many covariates cannot be very useful. It is our hope that, in the spirit of this paper, research will strive not toward bigger models of prediction but rather toward better ones.

## References

- [1] De Wilde, Philippe. 1997. *Neural Network Models: Theory and Projects*. Great Britain: Springer.
- [2] Neal, Radford M. 1996. *Bayesian Learning from Neural Networks*. New York: Springer.

- [3] Ripley, B. D. 1996. *Pattern Recognition and Neural Networks*. Great Britain: Cambridge.
- [4] Beneish, Messod D., Charles M. C. Lee, and Robin L. Tarpley. 2001. “Contextual Fundamental Analysis Through the Prediction of Extreme Returns.” *Review of Accounting Studies* 6:165–189.
- [5] King, Gary and Langche Zeng. 2001. “Logistic Regression in Rare Events Data.” *Political Analysis* 9(2):137–163.