Journal of Intelligent Learning Systems and Applications, 2019, 11, 33-63
http://www.scirp.org/journal/jilsa
ISSN Online: 2150-8410
ISSN Print: 2150-8402

Predicting Credit Card Transaction Fraud Using Machine Learning Algorithms

Jiaxin Gao[1], Zirui Zhou[2*], Jiangshan Ai[3], Bingxin Xia[4], Stephen Coggeshall[5]

[1]Hebei University of Economics and Business, Shijiazhuang, China
[2]China University of Political Science and Law, Beijing, China
[3]Wuhan Maple Leaf International School (High School), Wuhan, China
[4]Wuhan Jinde Education Consulting, Co., Ltd., Wuhan, China
[5]University of Southern California, Los Angeles, USA
Email: *luminav_zzr@163.com

How to cite this paper: Gao, J.X., Zhou, Z.R., Ai, J.S., Xia, B.X. and Coggeshall, S. (2019) Predicting Credit Card Transaction Fraud Using Machine Learning Algorithms. *Journal of Intelligent Learning Systems and Applications*, **11**, 33-63.
https://doi.org/10.4236/jilsa.2019.113003

Received: April 6, 2019
Accepted: August 11, 2019
Published: August 14, 2019

Abstract

Credit card fraud is a wide-ranging issue for financial institutions, involving theft and fraud committed using a payment card. In this paper, we explore the application of linear and nonlinear statistical modeling and machine learning models on real credit card transaction data. The models built are supervised fraud models that attempt to identify which transactions are most likely fraudulent. We discuss the processes of data exploration, data cleaning, variable creation, feature selection, model algorithms, and results. Five different supervised models are explored and compared including logistic regression, neural networks, random forest, boosted tree and support vector machines. The boosted tree model shows the best fraud detection result (FDR = 49.83%) for this particular data set. The resulting model can be utilized in a credit card fraud detection system. A similar model development process can be performed in related business domains such as insurance and telecommunications, to avoid or detect fraudulent activity.

Keywords

Credit Card Fraud, Machine Learning Algorithms, Logistic Regression, Neural Networks, Random Forest, Boosted Tree, Support Vector Machines

1. Introduction

Credit card fraud remains an important issue for theft and fraud committed using a payment card, such as a credit card or debit card. To combat this many fraud detection algorithms are widely used in industry [1] [2] [3] [4]. Card fraud can happen with the theft of the physical card as well as with the compromise of

the card, including skimming, breach, account takeover, that would otherwise look like a legitimate transaction. According to the Global Payments Report 2015 [5], the credit card is the highest-used payment method globally in 2014 compared to other methods such as an e-wallet and Bank Transfer. Along with the rise of credit card usage, the number of fraud cases has also been steadily increasing [6]. The rise in credit card fraud has a large impact on the financial industry. The global credit card fraud in 2015 reached a staggering USD 21.84 billion [7].

Financial institutions today typically develop custom fraud detection systems targeted to their own portfolios [8]. The data mining and machine learning techniques are vastly embraced to analyze patterns of normal and unusual behavior as well as individual transactions in order to flag likely fraud. Given the reality, the best cost-effective option is to tease out possible evidence of fraud from the available data using statistical algorithms [9]. Supervised models trained on labeled data examine all previous labeled transactions to mathematically determine how a typical fraudulent transaction looks and assigns a fraud probability score to each transaction [10]. Among the supervised algorithms typically used, the neural network is popular, and support vector machines (SVMs) have been applied, as well as decision trees and other models [3] [9] [11]-[21]. However, little attention has been devoted in the literature to some comparison of all the common algorithms, particularly using real data sets.

In this paper, we explore the application of various linear and nonlinear statistical modeling and machine learning models on credit card transaction data. The models built are supervised fraud models that attempt to identify which transactions are most likely fraudulent.

2. Description of Data

The data available for this research project are a collection of credit card transactions from a government agency located in Tennessee, U.S.A. The particular agency is not known.

The data consist of 96,753 credit card transactions during the year 2010, with 1059 labeled as fraud. The file contains the fields:

- Record: A unique identifier for each data record. This field also represents the time order;
- Cardnum: The account number for the transactions (we note that they are Mastercard transaction since the account numbers begin with the digits 54);
- Date: The date of the transaction. Month, day and year only (no time of day);
- Merchnum: A typically 12-digit merchant identification number;
- Merch Description: A brief text description field of the merchant, typically around 20 characters;
- Merch State: The state of the address for the merchant;
- Merch Zip: The zip code of the merchant;
- Transtype: A code denoting the type of transaction;

- Amount: The dollar amount of the transaction;
- Fraud: A label for the transaction to indicate whether or not it was a fraudulent transaction.

Table 1 shows summary information about all the fields. Only the Amount field is a numeric type field; the other fields are all categorical or text. The statistical magnitudes in the table were calculated with the outliers eliminated. Three fields have some missing values: Merchnum, Merch state, and Merch zip. It was noticed that the number of unique values of the Merch state field is 227, which is unexpected because the U.S. has only 50 states. Some of the values in this field might be from other countries, such as Canada and/or Mexico.

Below we show some further information about the data.

Figure 1 shows the number of transactions each month. We noticed the general upward trend through September, followed by a sharp drop in October. The monthly transactions are fewer in the last quarter of the year compared with other quarters. This is due to the government fiscal year which starts on October 1, and people tend to be more cautious with their money in the first few months of the new fiscal year.

Figure 2 shows the top 10 of the most frequently traded merchant descriptions. The total transaction frequency of the top 15 categories is 13,256, which is about 13.7% of the records. The top 200 categories account for 41% of the total records. In Table 1, we see that there are 13,126 kinds of merchants by this Merch description field, and 48.6% of the merchant descriptions only occurred once.

Figure 3 depicts the top 10 of the most frequently observed merchant states.

Table 1. Summary description of the data set.

Fields name	Type	Records that have a value	Percent populated	Mode	# Unique values	
Record	Index	96,753	100%			96,753
Cardnum	Categorical	96,753	100%	5142148452		1645
Date	Time	96,753	100%	2/28/10		365
Merchnum		93,378	96.5%	930090121224		13,091
Merch description		96,753	100%	GSA-FSS-ADV		13,126
Merch state	Categorical	95,558	98.8%	TN		227
Merch zip		92,097	95.2%	38118		4567
Transtype		96,753	100%	P		4
Amount	Numeric	96,753	100%	3.62	Mean 395.33* Max 30372.46* Min 0.01 Std 814.74*	34,909
Fraud	Categorical	96,753	100%	0		2

*Statistical magnitude without outliers.

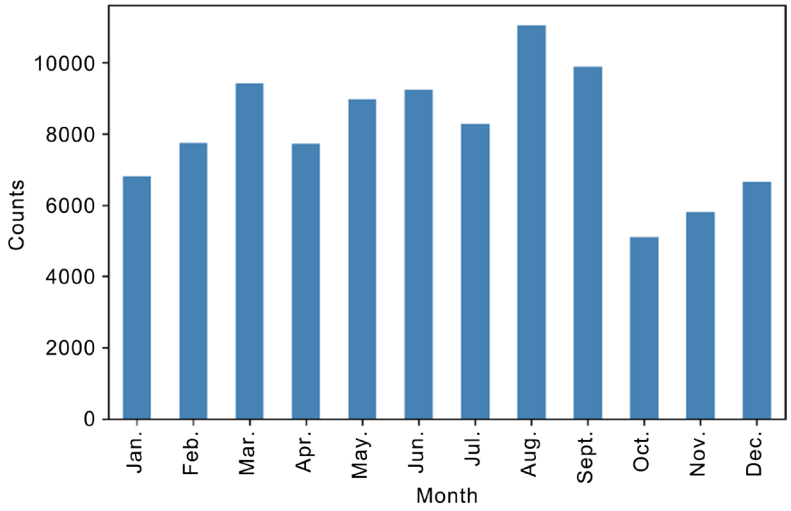

Figure 1. Monthly count of transactions shows seasonality.

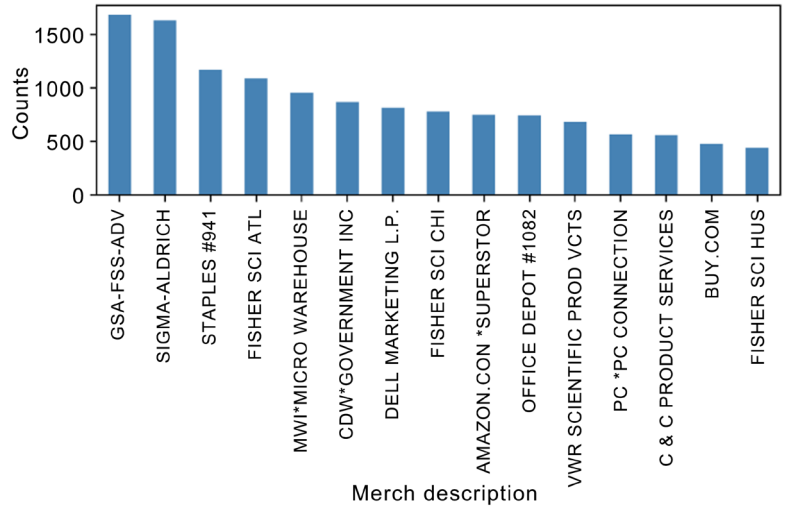

Figure 2. Most common merchant descriptions.

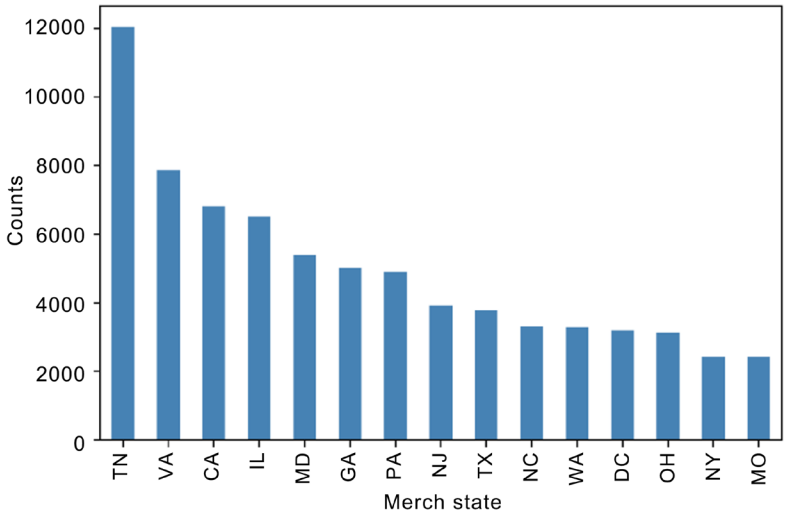

Figure 3. Most common states.

The most frequent state is TN which is about 12.6% of the whole transaction frequency, and is not surprising because that is where the facility is located. The total number of transactions in the top 15 states is 71,647, which is about 75% of the entire records. From Table 1 we see that there are 227 different states in the data. And we note that 168 of the state's field values are numbers rather than letters.

3. Data Cleaning

When we examine the field Amount, shown in the box plot distribution Figure 4, we see that there is one large outlier with the Amount value recorded as over 3 million dollars. After thoroughly checking that particular record, which is not labeled as fraud, we discover that it is an unusual but known transaction in Mexican pesos from a particular Mexican organization, and we thus exclude it from further analysis.

Table 2 shows the information about the fields with missing values. All of these three fields have a strong relationship with the field Merch description, but the same Merch description may also correspond to different values in the three above fields. These three fields also have a strong relationship with each other, so the mode of each field is used to fill in the missing fields.

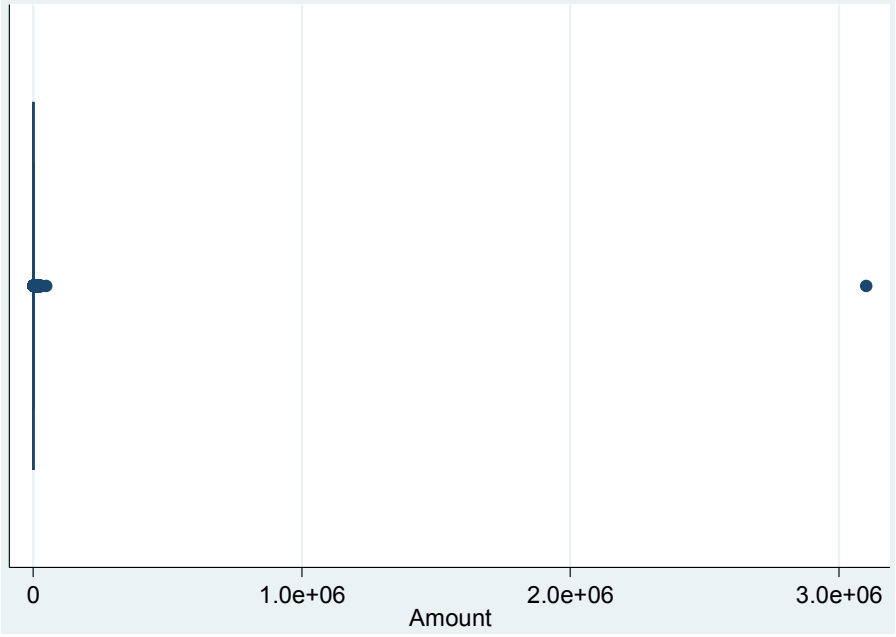

Figure 4. Box plot distribution of the Amount field.

Table 2. Fields with missing values.

Field	Records with missing values	Percentage (%)	Mode
Merchnum	3357	3.49	930090121224
Merch state	1195	1.24	TN
Merch zip	4565	4.81	38118

4. Variable Creation

Creating expert variables is a critical step before any model can be built. We examine the original fields from the raw data set, as shown in **Table 1**, and from these we create a large universe of candidate variables for our supervised models. At this stage, we want to create as many candidate variables as possible, and later we will use a variety of feature selection methods to reduce the universe of candidate variables to those that will be our final set of possible model inputs.

This step of creating variables, also called feature engineering, requires us to know as much as possible about the dynamics of the particular problem we are trying to solve. We want to do our best to create variables that in themselves contain as much encapsulated information of the signals of anomalous behavior that could be fraud. Thus we interview domain experts to fully understand as best as possible the different modes of fraud for this problem and try to build variables that will show the signals of these fraud modes.

This step of variable creation is arguably the most important step in machine learning. Encapsulating the problem dynamics as best as possible into these expert variables, prepares the data for the model in an as optimal way as possible. One should always do as much work upfront in this stage as possible in order to minimize the work required by your machine learning algorithm. It is true that theoretically all these nonlinear algorithms can discover these expert variable dynamics themselves, but it is generally difficult in a high dimensional space with sparse data sets. We note that 100,000 records are sparse data in anything above a handful of dimensions. Thus we work hard to create a universe of candidate variables that have encoded in them as best as possible as many indicators of potential fraud as we can think of. The logic behind these special expert variables is described below [22].

There are four general types of expert variables that have been built:

1) Amount variables. These variables are important because the amount spent can be immediately indicative of unusual activity. For example, if a person typically spends about 400 dollars over some time window and one day he spends 400,000 dollars in one transaction, it would be a signal of odd behavior. Therefore these amount-type variables are good candidates to detect potential frauds.

$$\left\{\begin{array}{l}\text{maximum}\\ \text{median}\\ \text{total}\\ \text{actual-average}\\ \text{actual-median}\\ \text{actual/average}\\ \text{actual/max}\\ \text{actual/total}\\ \text{actual/median}\end{array}\right\}\ \text{amount spent by/at this}\ \left\{\begin{array}{l}\text{cardnum}\\ \text{merchnum}\\ \text{cardnum on this merchnum}\\ \text{cardnum in this zip code}\\ \text{cardnum in this state}\end{array}\right\}\ \text{over the last}\ \left\{\begin{array}{l}\text{1 day}\\ \text{7 days}\\ \text{14 days}\\ \text{30 days}\end{array}\right\}$$

2) Frequency variables. Another example to illustrate the significance of this is

that if a person does 2 transactions per day usually, but one day he does 2000 transactions. That will be suspicious. So frequency variables are helpful to see if there is a big difference in a number of transactions.

$$\text{number of transactions with same} \begin{cases} \text{cardnum} \\ \text{merchnum} \\ \text{cardnum and merchnum} \\ \text{cardnum and zip code} \\ \text{cardnum and state} \end{cases}$$

$$\text{over the last} \begin{cases} \text{1 day} \\ \text{7 days} \\ \text{14 days} \\ \text{30 days} \end{cases}$$

3) Days since variables. One example is that if it has been 10 months since the last time a person uses his card, someone else might be using it in this case, so that it might be a fraud.

$$\text{Current transaction date minus date of most recent transaction with same} \begin{cases} \text{cardnum} \\ \text{merchnum} \\ \text{cardnum and merchnum} \\ \text{cardnum and zip code} \\ \text{cardnum and state} \end{cases}$$

4) Velocity variables. This measure shows how a person may do a large number of transactions over a day compared to his average transactions over a period of time. It is similar to the frequency variables.

$$\frac{\text{number of transactions with same} \begin{cases} \text{cardnum} \\ \text{merchnum} \end{cases} \text{over the last 1 day}}{\text{average daily number of transactions for this} \begin{cases} \text{cardnum} \\ \text{merchnum} \end{cases} \text{over the last} \begin{cases} \text{7 days} \\ \text{14 days} \\ \text{30 days} \end{cases}}$$

with some variations of characteristics we end up 200 variables of the first type, 20 variables of the second type, 5 variables of the third type and 12 variables of the fourth type. These 237 expert variables are listed in **Appendix**.

5. Feature Selection

A non intuitive fact of dimensionality is that all data points tend to be outliers as dimensionality increases. Another danger of high dimensionality is the exponentially growing need of data to find true nonlinearity rather than noise. These facts, generally called the curse of dimensionality, demand that one does feature selection before attempting to build models in general, but particularly for any nonlinear models. Feature selection is of great necessity through the whole building process of machine learning algorithm as it not only drastically decreases the dimensionality but also helps discover the particular variables that

carry the more useful information.

In general, there are three categories of feature selection methods: filter, wrapper, and embedded methods. The former two are applied in this research. A filter is a method applied to the data set without using a particular standard modeling algorithm. The wrapper method "wraps" a particular model around the feature selection process, for example, stepwise selection. Finally, an embedded method has the feature selection built directly into the modeling process, such as various tree methods or the use of regularization.

Here we first apply a filter method of feature selection to reduce the number of variables by about half. By calculating certain performance measures across every single variable individually and ranking the variables by these measures, a filter method can efficiently identify those with low performance. The particular filter measures we use for this fraud problem are a univariate Kolmogorov-Smirnov (KS) and the univariate fraud detection rate (FDR) at 3%. The Kolmogorov-Smirnov statistical test is a measure to quantitatively determine how much two distributions over the same independent variable are separated. This is a common test used for feature selection in a binary classification problem. Here we build the distribution of the two categories of records, one for each of the two dependent variables values, across each of the independent variables, one by one. The question we are asking is how well each independent variable by itself can differentiate the two classes. The equations for calculating the KS separation of two distributions are:

Kolmogorov-Smirnov Formula

$$KS = \max_{x} \int_{x_{\min}}^{x} \left[P_{good} - P_{bad} \right] \mathrm{d}x$$

$$KS = \max_{x} \sum_{x_{\min}}^{x} \left[P_{good} - P_{bad} \right]$$

where the first representation is for a continuous distribution and the second is what we use in practice for finite data sets.

For each candidate variable, we build the two distributions of the frauds and the non-frauds by this variable, and we calculate the KS separation between these two distributions. This measure explicitly tells how well each candidate variable by itself can separate the classes, and is, in essence, a univariate model.

Another measure of goodness we can calculate is the univariate fraud detection rate (FDR) for each candidate variable. The FDR is the measure of what percent of total frauds are caught at a particular percent of the population, assorted by the fraud score or, in this case, the candidate variable. For a particular candidate variable, we sort all the records by the value of this variable. We then proceed from the "top" of this list and add up the # frauds observed as we penetrate deeper into the list of records. We count the number of frauds observed at each of the percents of penetration of the data population, and discover, for example, that in this sorted list we might find that 10% of the frauds are contained in the top 3% of the records assorted by this particular candidate variable. This

FDR (10% fraud at 3% penetration) is a measure of how well the candidate variable by itself separates the frauds from the non-frauds.

A wrapper is a popular feature selection method to help determine which variables should be included in the machine learning model. It directly builds many models with different sets of variables, selecting them by examining the measure of goodness of the model itself. The most common wrapper method is stepwise selection, typically forward or backward selection. For forward selection, we first build univariate models, one for each of the n candidate variables, and examine the measure of goodness for each model. We select the variable x_i whose univariate model is best and then proceed by building n-1 two-variable models, x_i combined with all the remaining candidate variables, one by one. We select the best 2-variable model, now having identified x_i and another x_j as the strongest variables (x_j strong in conjunction with x_i). We continue this stepwise forward selection until we see negligible improvement when adding a new variable.

Backward selection is similar. Here we start with a single n-variable mode that uses all candidate variables. We then build n new models, each using only n-1 variables, where we have removed one of the n possible variables from each new model. The variable whose n-1 model decays the least is then discarded, and we continue by building n-1 2-variable models using all combinations of the remaining n-1 candidate variables. We continue this way until the removal of the next variable causes undesirable model performance degradation.

Both of these wrapper methods result in a rank-ordered list of the variables, sorted by importance in predicting the dependent variable. Note that this wrapper feature selection is done with a particularly selected modeling method wrapper, and while the rank-order importance is not strongly dependent on the choice of wrapper model method there may be some dependence on this choice.

Because so many models need to be built in stepwise feature selection, it is common to use a very fast modeling method as the wrapper, and one typically uses linear or logistic regression. We note that this common and expedient choice generally ignores the potential of variable interactions. In general, the choice of wrapper model method can be independent of the choice of model method for the final model. In this work, we used logistic regression, FDR at 3% and the forward selection method for the wrapper feature selection process.

We begin our feature selection process with the universe of 237 expert variables created, and first apply our filter methods to reduce the number of expert variables from 237 to 120. By combining the KS and FDR with equal weight, the filter reduces our variable set to about half of the variables, and then we apply the wrapper to reduce to the final 20 variables, still rank-ordered by importance. Figure 5 illustrates the stages of this two-step feature selection process.

Table 3 below lists the final 20 variables after feature selection.

6. Model Algorithms

We explore the use of a variety of supervised modeling methods, starting with a baseline logistic regression and then a number of nonlinear statistical/machine

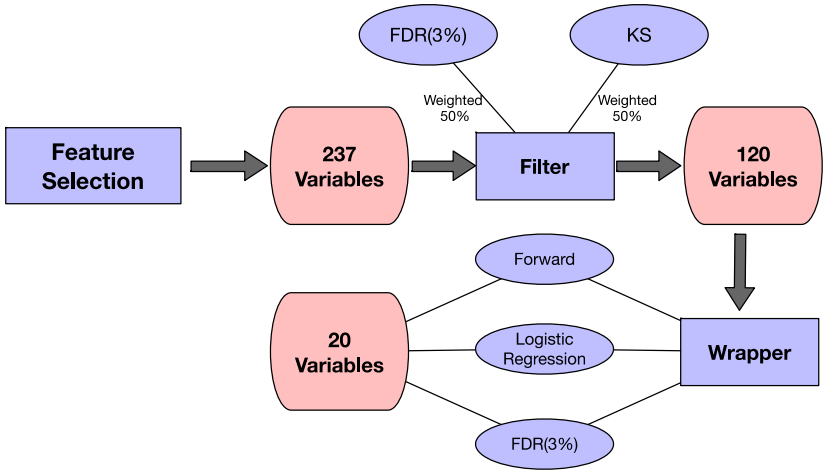

Figure 5. Feature selection process.

Table 3. The selected top 20 variables ranked by importance after feature selection.

Ranking	Variables
1	total amount by this cardnum at this merchnum in 7 days
2	max amount by this cardnum in 30 days
3	total amount by this cardnum in this zip code in 14 days
4	actual-median amount by this cardnum in this zip code in 30 days
5	total amount by this merchnum in 1 day
6	total amount by this merchnum in 14 days
7	max amount by this merchnum in 30 days
8	total amount by this merchnum in 7 days
9	total amount by this cardnum at this merchnum in 14 days
10	#transactions by this cardnum in 1 day
11	total amount by this cardnum at this merchnum in 1 day
12	actual/average amount by this cardnum in 30 days
13	#transactions by this cardnum in this state in 1 day
14	max amount by this cardnum in this state in 14 days
15	median amount by this cardnum in this zip code in 30 days
16	median amount by this cardnum in 30 days
17	total amount by this cardnum in this zip code in 30 days
18	actual-average amount by this cardnum in 30 days
19	actual-average amount by this cardnum in 7 days
20	actual/average amount by this merchnum in 30 days

learning methods. For each method, we describe the general principals of the algorithm and the parameter searches performed.

In this analysis we divide our data set into three parts: training, testing and out of time/validation. This third data set is chosen to be the final 2 months of

our data. We use the first 10 months of our data set to build our best models, randomly dividing the records into training and testing in multiple ways, and then we evaluate the model on this out of time data set. This gives us the most likely model performance that will be experienced when the model is implemented. Although we use this out of time evaluation multiple times during model tuning, it still provides a more realistic performance expectation measure than what the in-time testing data provides.

6.1. Logistic Regression

With 20 variables selected, logistic regression functions as a baseline model for its simplicity of parameters interpretation. We note that based on its simplicity and generally fine performance, logistic regression is frequently the method of choice for many real-world business classification problems. For this study, the first step is to build a baseline logistic regression using all variables from our feature selection process. Table 4 below is a summary of the main parameters of this logistic regression model. Each variable x_i is the variable given in Table 3.

Note that x_3 and x_{16} both have p values higher than 0.05, and therefore should be excluded since they are not statistically significant. Thus the new version of logistic regression on the remaining 18 variables (removing x_3 and x_{16}) is shown in Table 5 summary below.

Table 4. First logistic regression output on the training data.

Optimization terminated successfully.						
Current function value: 0.026521						
Iterations 10						
Results: Logit						
Model:	Logit			Pseudo R-squared:		0.539
Dependent variable:	y			AIC:		3382.435
Date:	2019/2/1 20:01			BIC:		3572.495
No. observations:	62976			Log-Likelihood:		−1670.2
Df Model:	20			LL-Null:		−3619.4
Df residuals:	62955			LLR p-value:		0.0000
Converged:	1.0000			Scale:		1.0000
No. iterations:	10.0000					
	Coef.	Std. Err.	z	P > |z|	[0.025	0.975]
Const	−6.3774	0.0952	−66.9633	0.0000	−6.5640	−6.1907
x_1	−0.3720	0.0991	−3.7524	0.0002	−0.5664	−0.1777
x_2	0.3912	0.0553	7.0703	0.0000	0.2828	0.4997
x_3	−0.0651	0.0853	−0.7637	0.4450	−0.2322	0.1020
x_4	−0.5262	0.0736	−7.1466	0.0000	−0.6705	−0.3819
x_5	0.6035	0.0385	15.6722	0.0000	0.5280	0.6789

Continued

x_6	−1.8396	0.1679	−10.9593	0.0000	−2.1686	−1.5106
x_7	−0.8686	0.0907	−9.5745	0.0000	−1.0465	−0.6908
x_8	1.6247	0.1464	11.0981	0.0000	1.3378	1.9117
x_9	0.8138	0.1118	7.2775	0.0000	0.5947	1.0330
x_{10}	0.9733	0.0953	10.2176	0.0000	0.7866	1.1600
x_{11}	−0.4629	0.0645	−7.1806	0.0000	−0.5892	−0.3365
x_{12}	0.2565	0.0439	5.8379	0.0000	0.1704	0.3426
x_{13}	−0.5733	0.1005	−5.7035	0.0000	−0.7703	−0.3763
x_{14}	0.4198	0.0527	7.9630	0.0000	0.3164	0.5231
x_{15}	−0.1589	0.0636	−2.5002	0.0124	−0.2835	−0.0343
x_{16}	0.0146	0.0536	0.2721	0.7856	−0.0905	0.1197
x_{17}	0.3298	0.0405	8.1483	0.0000	0.2505	0.4091
x_{18}	0.7775	0.0960	8.0953	0.0000	0.5893	0.9658
x_{19}	−0.3247	0.0478	−6.7955	0.0000	−0.4183	−0.2310
x_{20}	0.2963	0.0300	9.8665	0.0000	0.2374	0.3551

Table 5. Final logistic regression output on the training data.

Optimization terminated successfully.						
Current function value: 0.026716						
Iterations 10						
Results: Logit						
Model:	Logit			Pseudo R-squared:		0.538
Dependent variable:	y			AIC:		3402.9056
Date:	2019/2/1 19:57			BIC:		3574.8652
No. observations:	62976			Log-Likelihood:		−1682.5
Df Model:	18			LL-Null:		−3642.2
Df residuals:	62957			LLR p-value:		0.0000
Converged:	1.0000			Scale:		1.0000
No. iterations:	10.0000					
	Coef.	Std.Err.	z	P > |z|	[0.025	0.975]
Const	−6.3268	0.0932	−67.8687	0.0000	−6.5096	−6.1441
x_1	−0.3722	0.1000	−3.7566	0.0002	−0.5714	−0.1796
x_2	0.4580	0.0533	8.5863	0.0000	0.3534	0.5625
x_3	−0.5297	0.0607	−8.7208	0.0000	−0.6488	−0.4107
x_4	0.5835	0.0386	15.1147	0.0000	0.5078	0.6591
x_5	−1.7827	0.1667	−10.6910	0.0000	−2.1095	−1.4559
x_6	−0.9201	0.0903	−10.1885	0.0000	−1.0970	−0.7431
x_7	1.6306	0.1491	10.9381	0.0000	1.3384	1.9228

Continued

x_8	0.7779	0.0893	8.7147	0.0000	0.6030	0.9529
x_9	0.9498	0.0906	10.4809	0.0000	0.7722	1.1274
x_{10}	−0.4543	0.0652	−6.9706	0.0000	−0.5820	−0.3265
x_{11}	0.2262	0.0439	5.1566	0.0000	0.1402	0.3122
x_{12}	−0.5552	0.0966	−5.7447	0.0000	−0.7446	−0.3658
x_{13}	0.3950	0.0516	7.6543	0.0000	0.2939	0.4962
x_{14}	−0.1463	0.0581	−2.5207	0.0117	−0.2601	−0.0326
x_{15}	0.2931	0.0405	7.2453	0.0000	0.2138	0.3724
x_{16}	0.7826	0.0943	8.2965	0.0000	0.5977	0.9674
x_{17}	−0.3139	0.0494	−6.3490	0.0000	−0.4108	−0.2170
x_{18}	0.2648	0.0307	8.6278	0.0000	0.2047	0.3250

To optimize the logistic regression the number of variables chosen is an adjustable factor in the tuning process. Models with different numbers of variables are compared in Table 6 below. Note that each experiment of a different number of variables is achieved by removing the last several variables since they are ranked by importance.

Table 6 shows the results of FDR (at 3%) on the out of time data for logistic regression using different numbers of variables, always selected in the rank order shown in Table 3. The nonmonotonic nature of the results is simply a statistical variation due to the sparsity of records.

6.2. Artificial Neural Network

An artificial neural network (ANN) is an algorithm that shares simulative structure of the neural network of human brain. As shown in Figure 6. There are an input layer, hidden layers and an output layer in the overall network architecture. Each of these layers contains nodes or neurons, which are gathering locations for the receipt and transmission of numerical signals from the previous to the next layer of the neural net. Each neuron embedded in the structure receives a signal from the nodes in the previous layer, applies a transfer function to the signal, and then outputs a new signal to the nodes in the next layer. The signal received from the previous layer is in general a linear combination of the outputs of the previous layer nodes. This combined linear combination signal, received by the node, is then passed through a transfer function, typically a sigmoid or logit function. Other transfer functions are also used, but the sigmoid/logit is the most common.

In this work, we use this sigmoid/logistic nonlinear transfer function, with the equation below.

$$S(x)=\frac{1}{1+\mathrm{e}^{-x}}$$

$$S'(x)=\frac{\mathrm{e}^{-x}}{\left(1+\mathrm{e}^{-x}\right)^2}=S(x)\left(1-S(x)\right)$$

Table 6. Logistic regression performance for different number of variables.

OOT FDR for logistic regression	
Number of variables	%OOT FDR (at 3%)
5	44.30
6	45.25
8	44.53
10	31.84
15	34.47
20	39.22

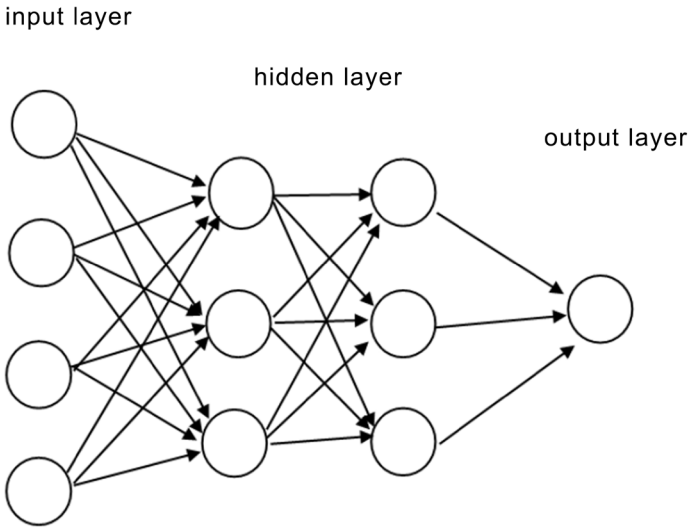

Figure 6. Neural net architecture[1].

The variable x in this equation is the linear combination of the weighted signals received from the nodes in the previous layer.

During the parameter tuning process, the key parameters that mainly define the structure of the ANN are number of variables/inputs, hidden layer sizes (number of layers & number of nodes in each layer), a regularization weight alpha and the learning rate, all of which become the searchable parameters in our tuning experiments. Table 7 below shows the parameters settings of the optimal ANN after tuning.

We note that, even though this was our best ANN model it is likely not the best possible. With only one node in a single hidden layer and the logit transform function, it is mathematically equivalent to a logistic regression.

6.3. Support Vector Machine

A Support Vector Machine (SVM) is a wildly applied binary classifier for supervised machine learning problems. The main characteristics of the SVM algorithm are:

[1]The figure is quoted from http://image.baidu.com.

Table 7. Final parameters for the Artificial Neural Net model.

Main parameters of artificial neural net			
Activation:	Logistic	Solver:	Sgd
Alpha:	0.01	Learning rate:	Adaptive
Hidden layers sizes:	(1)*	Bach size:	Auto
Max iter:	200	Number of variables:	6

*layer: 1 nodes: 1.

- Expand the dimensionality;
- Look for a linear classifier separation surface in this higher dimensional space.

It is counterintuitive to expand the dimensionality of a nonlinear machine learning algorithm, since as we argued previously, dimensionality is in a sense the enemy of nonlinear modeling. Expanding the dimensionality can provide new variables, such as the many varieties of the products and higher order powers of the original variables, where a linear separation in this complex higher dimensional space might be a good separation surface. But adding these dimensions is against the principals recognized by the curse of dimensionality.

The SVM algorithm achieves this balance by not using explicit new variables in this extended dimensionality, but introduces "virtual" new variables through the use of a kernel trick. It is observed that as the algorithm searches for the best linear separator in the space, all that is needed is some measure of distance. The kernel is a generalized measure of distance and can be employed in this virtual higher dimension without explicitly calculating new variables or explicitly expanding the space.

Another important characteristic of the SVM algorithm is the use of a margin concept, where the best linear classifier is selected not just on the basis of the classification error but also considering the best spread around the data, tilting the surface so that this margin of classification/misclassification is as optimal as possible. This is shown in **Figure 7**.

The choice of a radial basis function (RBF) kernel function is used in this study. The number of variables, C and gamma, are key parameters of the SVM algorithm with the RBF kernel. C is a weight for each of the classes, set to one here, and gamma is a scale factor for the width of the radial basis functions. **Table 8** below shows the final settings of parameters of the SVM after tuning.

6.4. Random Forest

A decision tree algorithm is one where the space of data is separated into a number of distinct boxes that, combined, cover all the possible space. It is built by doing iterative cuts of the space, typically dividing an existing box into two "child" boxes, which is thus called a binary tree. At each iteration, the algorithm considers a box of data and examines what might be the best way to split it into two separate boxes where some criterion is optimal. The splitting criterion for a binary classification problem is typically an impurity measure, where the split

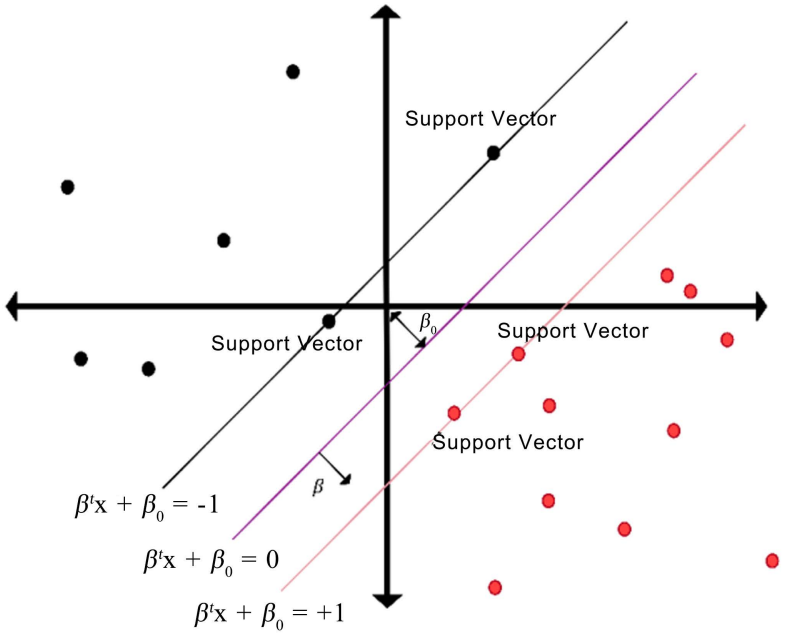

Figure 7. Support vector machine separation surface[2].

Table 8. Parameter settings for the SVM model.

Main parameters of support vector machine			
Kernel:	rbf	Number of variables:	6
C:	1	Gamma:	1

attempts to separate into two child boxes of the greatest type purity, meaning, the two classes are separated as well as possible. The splitting algorithm continues, considering each child box as a new parent box, and continues to split until a stopping criterion is reached.

Decision trees are an early and ubiquitous machine learning, nonlinear statistical modeling methodology. Used extensively decades ago they are known to have serious flaws that are easy to succumb to. The primary deficiency is the tendency to overfit, followed by instability and fragility. The structure of a tree can change substantially depending on near-random choices of the first few splitting locations, which can easily change with small changes in the fitting data set. These deficiencies have been largely overcome through more modern algorithms that take advantage of the good properties of trees and greatly avoid these known pitfalls. The use of multiple trees rather than a single complex tree substantially removes many of the problems. Two widely-used architectures that use multiple trees are random forests and boosted trees.

A random forest, shown in **Figure 8**, is a large collection of decision trees, where the final output is a committee choice across many individual trees. The principle of random forest is that many trees are built, and each uses a randomly-chosen subset of features. All the results are then combined, typically by

[2]The figure is quoted from http://image.baidu.com.

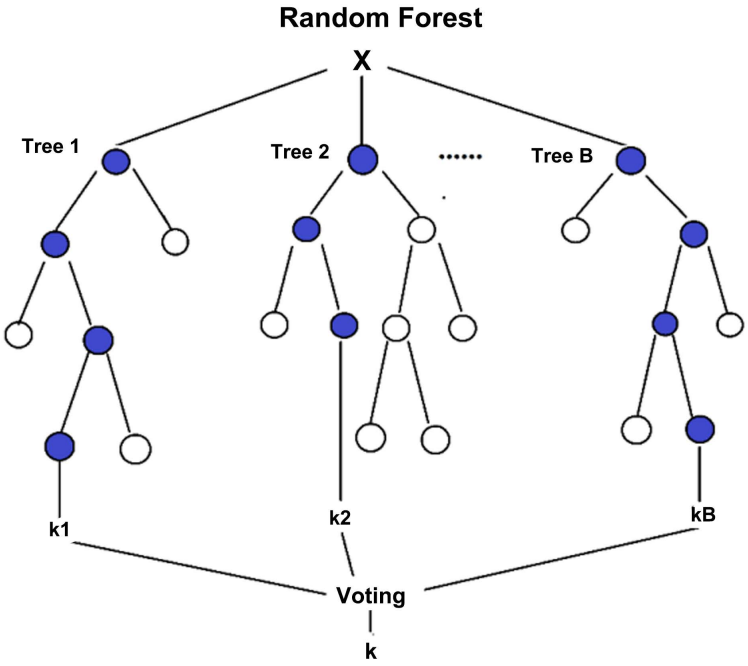

Figure 8. Random forest architecture.

averaging or voting. A random forest, as compared to a single decision tree, can get a more accurate prediction. Further, since the tree depths are usually smaller and many trees are combined, overfitting issues are largely avoided. While tuning the parameters it is important to know what each parameter means. There are 5 important parameters that can be tuned for the random forest algorithm selected:

- max_features: Maximum number of features the random forest can try in each individual tree;
- max_depth: Maximum depth of each tree;
- min_samples_leaf: The minimum number of samples required to split an internal node;
- n_estimator: Number of trees to be built.

After multiple trials, the setting of parameters was finalized as is shown in Table 9 below.

6.5. Boosted Tree

Boosted trees or gradient boosting trees is a type of supervised model that again uses a collection of decision trees, but constructed in a very different way from the random forest. As shown in Figure 9, the boosting process is an iterative approximation process, where we incrementally add more trees, in a fashion similar fashion to a Taylor series, each adding slightly more predictive power to the whole.

This series of "weak learners" has several important characteristics. First, each tree is a very simple tree, limited in depth and variables. The first tree in the series makes a very crude fit to the target output. The error in this fit is calculated

Table 9. Parameter settings for the random forest model.

Final parameter choices of the random forest			
Max depth:	5	Max features:	6
Min sample leaf:	2	Min sample split:	10
Estimator:	400	Number of variables:	8

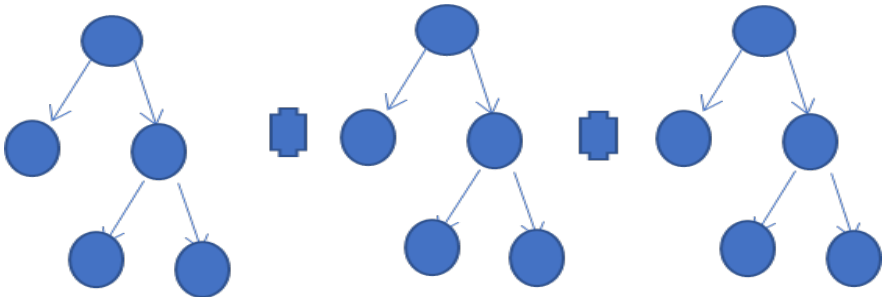

Figure 9. Boosted tree architecture.

for each record, and this error becomes a weighting factor for the next training iteration. Thus as this series of weak learners continues, each next additive tree tends to focus its learning objective on the records for which there is the largest error so far. The use of the latest error as the weights for the next training iteration is what gives the algorithm the name boosting, and indeed any weak learner can be used in a boosting algorithm.

Boosted tree models have a set of parameters that can be tuned to improve the quality of the series of weak learners. Some important parameters were explored, specifically the parameters.

Max_depth: Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit.

Learning_rate: Step size shrinkage used for the next iterative tree and can help to prevent overfitting. Empirically it has been found that using small learning rates yields dramatic improvements in the model's generalization ability over gradient boosting without shrinking.

Scale_pos_weight: Control the balance of positive and negative class weights, useful for unbalanced classes.

Min_child_weight: Minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weights less than min_child_weight, then the building process will give up further partitioning.

Table 10 below illustrates the final settings of these parameters for the boosted tree after tuning.

7. Model Results

Table 11 below summarizes the FDR at 3% for each model.

The ultimate purpose of this research is to find a relatively robust model which

Table 10. Parameter settings for the boosted tree model.

Main parameters of boosted tree			
Learning rate:	0.01	Number of trees:	300
Max depth:	5	Number of variables:	20

Table 11. Model results on the card transaction data on the training, testing and out of time validation data sets. These numbers are averages over 10 runs for each data set.

Model	FDR (at 3%)		
	Train (%)	Test (%)	OOT (%)
Logistic regression	72.85	72.96	45.25
Artificial neural net	72.65	68.95	44.86
Support vector machine	90.12	82.29	47.54
Random forest	80.97	79.58	45.42
Boosted tree	89.86	90.08	49.83

can handle the real-time data flow required for a credit card transaction fraud model. The out of time (OOT) data set is set aside, separated from the training/testing data, to simulate how the model will perform when implemented. Thus, the key measure to determine which nonlinear model to deploy is naturally the FDR performance on OOT data. Obviously, the boosted tree (FDR = 49.83%) slightly outperformed the next best model (SVM, FDR = 47.54%) and become our best choice for our nonlinear model.

Having settled on the boosted tree as our final model we then examined the FDR at different population cutoff locations, separately on training set, testing set and OOT data set. Tables 12-14 below illustrate these results. Here the FPR is the false positive ratio, the number of incorrectly flagged non-frauds divided by the correctly caught frauds, another common measure of model goodness.

These three tables show the model performance statistics of the three data sets (Table 12 of training data set, Table 13 of testing data set, Table 14 of OOT data set), and the OOT data set (Table 14) shows our best guess of how the model will perform when implemented. In this table, we see that the model pushes the majority of the fraud records to the top bins, which is what is desired. The bin statistics tell us what is happening in each population percentile bin and the cumulative statistics tell us what would happen if we were to select any particular percentage as a score cutoff. We see in the cumulative statistics the FDR of the OOT data set at 3%, which means that if we set the score cutoff at 3% we expect to catch about 50.84% percent of all the fraudulent transaction attempts. A score cutoff of 3% means that the business will reject the top 3% of transaction volume as sorted by the fraud score. Note that the FDR at 3% for each data set here are different from those in Table 11 because Table 11 reflects average FDR for over 10 runs for each data set, while FDR here is generated over one run when building boosted tree.

Table 12. Training results.

Training	#Records	#Goods	#Bads	Fraud rate
	62,976	62,339	637	0.0101

	Bin statistics						Cumulative statistics					
Population bin%	#Records	#Goods	#Bads	%Goods	%Bads	Total #records records	Cumulative good	Cumulative bad	%Good	%Bad (FDR)	KS	FPR
1	630	90	540	14.3	85.7	630	90	540	0.14	84.77	84.63	0.1667
2	630	601	29	95.4	4.6	1260	691	569	1.11	89.32	88.22	1.2144
3	630	628	2	99.7	0.3	1890	1319	571	2.12	89.64	87.52	2.3100
4	630	627	3	99.5	0.5	2520	1946	574	3.12	90.11	86.99	3.3902
5	630	629	1	99.8	0.2	3150	2575	575	4.13	90.27	86.14	4.4783
6	630	630	0	100.0	0.0	3780	3205	575	5.14	90.27	85.13	5.5739
7	630	628	2	99.7	0.3	4410	3833	577	6.15	90.58	84.43	6.6430
8	630	627	3	99.5	0.5	5040	4460	580	7.15	91.05	83.90	7.6897
9	630	630	0	100.0	0.0	5670	5090	580	8.17	91.05	82.89	8.7759
10	630	618	12	98.1	1.9	6300	5708	592	9.16	92.94	83.78	9.6419
11	630	624	6	99.0	1.0	6930	6332	598	10.16	93.88	83.72	10.5886
12	630	625	5	99.2	0.8	7560	6957	603	11.16	94.66	83.50	11.5373
13	630	630	0	100.0	0.0	8190	7587	603	12.17	94.66	82.49	12.5821
14	630	630	0	100.0	0.0	8820	8217	603	13.18	94.66	81.48	13.6269
15	630	630	0	100.0	0.0	9450	8847	603	14.19	94.66	80.47	14.6716
16	630	627	3	99.5	0.5	10,080	9474	606	15.20	95.13	79.94	15.6337
17	630	630	0	100.0	0.0	10,710	10,104	606	16.21	95.13	78.93	16.6733
18	630	630	0	100.0	0.0	11,340	10,734	606	17.22	95.13	77.91	17.7129
19	630	630	0	100.0	0.0	11,970	11,364	606	18.23	95.13	76.90	18.7525
20	630	629	1	99.8	0.2	12,600	11,993	607	19.24	95.29	76.05	19.7578

Table 13. Testing results.

Testing	#Records	#Goods	#Bads	Fraud rate
	20,993	20,750	243	0.0116

	Bin statistics						Cumulative statistics					
Population bin%	#Records	#Goods	#Bads	%Goods	%Bads	Total #records records	Cumulative good	Cumulative bad	%Good	%Bad (FDR)	KS	FPR
1	210	20	190	9.5	90.5	210	20	190	0.10	78.19	78.09	0.1053
2	210	183	27	87.1	12.9	420	203	217	0.98	89.30	88.32	0.9355
3	210	208	2	99.0	1.0	630	411	219	1.98	90.12	88.14	1.8767
4	210	210	0	100.0	0.0	840	621	219	2.99	90.12	87.13	2.8356
5	210	209	1	99.5	0.5	1050	830	220	4.00	90.53	86.53	3.7727

Continued

6	210	210	0	100.0	0.0	1260	1040	220	5.01	90.53	85.52	4.7273
7	210	208	2	99.0	1.0	1470	1248	222	6.01	91.36	85.34	5.6216
8	210	209	1	99.5	0.5	1680	1457	223	7.02	91.77	84.75	6.5336
9	210	210	0	100.0	0.0	1890	1667	223	8.03	91.77	83.74	7.4753
10	210	209	1	99.5	0.5	2100	1876	224	9.04	92.18	83.14	8.3750
11	210	206	4	98.1	1.9	2310	2082	228	10.03	93.83	83.79	9.1316
12	210	210	0	100.0	0.0	2520	2292	228	11.05	93.83	82.78	10.0526
13	210	210	0	100.0	0.0	2730	2502	228	12.06	93.83	81.77	10.9737
14	210	210	0	100.0	0.0	2940	2712	228	13.07	93.83	80.76	11.8947
15	210	209	1	99.5	0.5	3150	2921	229	14.08	94.24	80.16	12.7555
16	210	209	1	99.5	0.5	3360	3130	230	15.08	94.65	79.57	13.6087
17	210	209	1	99.5	0.5	3570	3339	231	16.09	95.06	78.97	14.4545
18	210	210	0	100.0	0.0	3780	3549	231	17.10	95.06	77.96	15.3636
19	210	209	1	99.5	0.5	3990	3758	232	18.11	95.47	77.36	16.1983
20	210	210	0	100.0	0.0	4200	3968	232	19.12	95.47	76.35	17.1034

Table 14. OOT results.

	#Records	**#Goods**	**#Bads**	**Fraud rate**
Out of time	**12,427**	**12,248**	**179**	**0.0144**

	Bin statistics					**Cumulative statistics**						
Population bin%	**#Records**	**#Goods**	**#Bads**	**%Goods**	**%Bads**	**Total #records**	**Cumulative good**	**Cumulative bad**	**%Good**	**%Bad (FDR)**	**KS**	**FPR**
1	124	81	43	65.3	34.7	124	81	43	0.66	24.02	23.36	1.8837
2	124	81	43	65.3	34.7	248	162	86	1.32	48.04	46.72	1.8837
3	124	119	5	96.0	4.0	372	281	91	2.29	50.84	48.54	3.0879
4	124	122	2	98.4	1.6	496	403	93	3.29	51.96	48.66	4.3333
5	124	123	1	99.2	0.8	620	526	94	4.29	52.51	48.22	5.5957
6	124	124	0	100.0	0.0	744	650	94	5.31	52.51	47.21	6.9149
7	124	103	21	83.1	16.9	868	753	115	6.15	64.25	58.10	6.5478
8	124	109	15	87.9	12.1	992	862	130	7.04	72.63	65.59	6.6308
9	124	124	0	100.0	0.0	1116	986	130	8.05	72.63	64.58	7.5846
10	124	123	1	99.2	0.8	1240	1109	131	9.05	73.18	64.13	8.4656
11	124	122	2	98.4	1.6	1364	1231	133	10.05	74.30	64.25	9.2556
12	124	124	0	100.0	0.0	1488	1355	133	11.06	74.30	63.24	10.1880
13	124	116	8	93.5	6.5	1612	1471	141	12.01	78.77	66.76	10.4326
14	124	122	2	98.4	1.6	1736	1593	143	13.01	79.89	66.88	11.1399
15	124	123	1	99.2	0.8	1860	1716	144	14.01	80.45	66.44	11.9167
16	124	123	1	99.2	0.8	1984	1839	145	15.01	81.01	65.99	12.6828
17	124	124	0	100.0	0.0	2108	1963	145	16.03	81.01	64.98	13.5379
18	124	124	0	100.0	0.0	2232	2087	145	17.04	81.01	63.97	14.3931
19	124	122	2	98.4	1.6	2356	2209	147	18.04	82.12	64.09	15.0272
20	124	122	2	98.4	1.6	2480	2331	149	19.03	83.24	64.21	15.6443

8. Conclusions and Further Work

In this paper, we explore the application of linear and nonlinear statistical and machine learning models on credit card transaction data. The models we build are supervised fraud models that attempt to identify which transactions are most likely fraudulent.

As we would expect, the nonlinear models slightly outperform the linear model, except for the artificial neural network. We believe this underperformance is due to 2 reasons. First, we recognize that we have likely not sufficiently tuned the neural net model and improvement can be found with a different set of model parameters. Second, we note that the data set is substantially limited, and only has 179 labeled fraud events in this OOT data set. The results of any model, particularly a nonlinear one, can be volatile and sensitive to the statistical aberrations and the variation of model parameters. The boosted tree model performs the best and can detect about half of the fraud attempts within only the top 3% data that was sorted as suspicious by the fraud algorithm score.

The resulting model can be utilized in a credit card fraud detection system. We note that similar model development process can be performed in related business domains such as insurance and telecommunications, to avoid or detect fraudulent activity.

We also found that we can use the scores of statistical significance in the logistic regression models as the criteria for selecting variables: high statistical significance score or the low p-value means strong correlation between fraud and related variable. It was found that the logistic regression model works quite well, once sufficiently good expert variables are constructed, which is also seen quite often in practice. What is observed in practice is that well-designed linear models are difficult to beat even with sophisticated nonlinear algorithms. The most important step is the construction of good expert variables that encode the signals of the problem dynamics as much as possible into clever variables. It is for this reason that linear and logistic regression models are so prevalent in the business field.

More data with more fields (for example, adding point of sale information, time of day, or other cardholder or merchant information) would certainly allow model performance improvements. Further model parameter tuning would also provide improvements to any of the models.

Acknowledgements

The authors would like to thank the Torhea Online Education Research Program as sponsors and hosts for this research project.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Hastie, T., Tibshirani, R. and Friedman, J. (2009) Boosting and Additive Trees. In:

The Elements of Statistical Learning, Springer, New York, 337-387.

[2] Jensen, D. (1997) Prospective Assessment of AI Technologies for Fraud Detection: A Case Study. In: *The AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*, AAAI Press, Palo Alto, CA, 34-38.

[3] Randhawa, K., Loo, C.K., Seera, M., Lim, C.P. and Nandi, A.K. (2018) Credit Card Fraud Detection Using AdaBoost and Majority Voting. *IEEE Access*, **6**, 14277-14284. https://doi.org/10.1109/ACCESS.2018.2806420

[4] Ryan, J., Lin, M.-J. and Miikkulainen, R. (1998) Intrusion Detection with Neural Networks. In: *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, MA, 943-949.

[5] Worldpay (2015) Global Payments Report 2015. http://offers.worldpayglobal.com/rs/850-JOA-856/images/GlobalPaymentsReportNov2015

[6] HSN Consultants (2016) The Nilson Report. https://www.nilsonreport.com/upload/content_promo/The_Nilson_Report_10-17-2016.pdf

[7] Stolfo, S., Fan, D.W., Lee, W., Prodromidis, A. and Chan, P. (1997) Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results. *AAAI-97 Workshop on Fraud Detection and Risk Management*, Providence, RI, 27-28 July 1997, 83-90.

[8] Wang, S. (2010) A Comprehensive Survey of Data Mining-Based Accounting-Fraud Detection Research. 2010 *International Conference on Intelligent Computation Technology and Automation*, Changsha, 11-12 May 2010, 50-53. https://doi.org/10.1109/ICICTA.2010.831

[9] Sherman, E. (2002) Fighting Web Fraud. *Newsweek*, **139**, 32B.

[10] Green, B.P. and Choi, J.H. (1997) Assessing the Risk of Management Fraud through Neural Network Technology. *Auditing*, **16**, 14-28.

[11] Albashrawi, M. (2016) Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015. *Journal of Data Science*, **14**, 553-569.

[12] Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S. and Bontempi, G. (2014) Learned Lessons in Credit Card Fraud Detection from a Practitioner Perspective. *Expert Systems with Applications*, **41**, 4915-4928. https://doi.org/10.1016/j.eswa.2014.02.026

[13] Estévez, P.A., Held, C.M. and Perez, C.A. (2006) Subscription Fraud Prevention in Telecommunications Using Fuzzy Rules and Neural Networks. *Expert Systems with Applications*, **31**, 337-344. https://doi.org/10.1016/j.eswa.2005.09.028

[14] Fawcett, T. and Provost, F. (1997) Combining Data Mining and Machine Learning for Effective Fraud Detection. *Proceedings of AI Approaches to Fraud Detection and Risk Management*, 14-19.

[15] Kumari, P. and Mishra, S.P. (2019) Analysis of Credit Card Fraud Detection Using Fusion Classifiers. In: Behera, H., Nayak, J., Naik, B. and Abraham, A., Eds., *Computational Intelligence in Data Mining. Advances in Intelligent Systems and Computing*, Springer, Singapore, 111-122. https://doi.org/10.1007/978-981-10-8055-5_11

[16] Navneet Jain, V.K. (2018) Credit Card Fraud Detection Using Recurrent Attributes. *International Advanced Research Journal in Science, Engineering and Technology*, **5**, 43-47.

[17] Sabau, A.S. (2012) Survey of Clustering Based Financial Fraud Detection Research.

Informatica Economica, **16**, 110.

[18] Vardhani, P.R., Priyadarshini, Y.I. and Narasimhulu, Y. (2019) CNN Data Mining Algorithm for Detecting Credit Card Fraud Soft Computing and Medical Bioinformatics. Springer, Singapore, 85-93. https://doi.org/10.1007/978-981-13-0059-2_10

[19] Wheeler, R. and Aitken, S. (2000) Multiple Algorithms for Fraud Detection. In: Ellis, R., Moulton, M. and Coenen, F., Eds., *Applications and Innovations in Intelligent Systems VII*, Springer, London, 219-231. https://doi.org/10.1007/978-1-4471-0465-0_14

[20] Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S. and Jiang, C. (2018) Random Forest for Credit Card Fraud Detection. 2018 IEEE 15*th International Conference on Networking*, *Sensing and Control*, Zhuhai, 27-29 March 2018, 1-6. https://doi.org/10.1109/ICNSC.2018.8361343

[21] Yee, O.S., Sagadevan, S. and Malim, N.H.A.H. (2018) Credit Card Fraud Detection Using Machine Learning as Data Mining Technique. *Journal of Telecommunication*, *Electronic and Computer Engineering*, **10**, 23-27.

[22] Gopinathan, K.M., Biafore, L.S., Ferguson, W.M., Lazarus, M.A., Pathria, A.K. and Jost, A. (1998) Fraud Detection Using Predictive Modeling. Google Patents.

Appendix: List of 237 Candidate Expert Variables

#transactions with same cardnum in 1 day/average daily #transactions for this cardnum in 7 days

#transactions with same cardnum in 1 day/average daily #transactions for this cardnum in 14 days

#transactions with same merchnum in 1 day/average daily #transactions for this merchnum in 7 days

#transactions with same merchnum in 1 day/average daily #transactions for this merchnum in 14 days

#transactions with same merchnum in 1 day/average daily #transactions for this merchnum in 30 days

#transactions with same cardnum in 1 day/average daily #transactions for this cardnum in 30 days

maximum amount by this cardnum in 30 days

total amount by this cardnum in 30 days

median amount by this cardnum in 30 days

median amount by this merchnum in 30 days

median amount by this cardnum at this merchnum in 30 days

median amount by this cardnum in this zip code in 30 days

median amount by this cardnum in this state in 30 days

total amount by this merchnum in 30 days

total amount by this cardnum at this merchnum in 30 days

total amount by this cardnum in this zip code in 30 days

total amount by this cardnum in this state in 30 days

maximum amount by this cardnum in this state in 30 days

maximum amount by this cardnum in this zip code in 30 days

maximum amount by this cardnum at this merchnum in 30 days

maximum amount by this merchnum in 30 days

average amount by this merchnum in 30 days

average amount by this cardnum at this merchnum in 30 days

average amount by this cardnum in this zip code in 30 days

average amount by this cardnum in this state in 30 days

average amount by this cardnum in 30 days

#transactions with same cardnum in 30 days

#transactions with same merchnum in 30 days

#transactions with same cardnum and merchnum in 30 days

#transactions with same cardnum and zip code in 30 days

#transactions with same cardnum and state in 30 days

#transactions with same cardnum in 1 day/30 days

#transactions with same merchnum in 1 day/30 days

actual-average amount by this cardnum in 30 days

actual by this merchnumedian amount by this cardnum in 30 days

actual/average amount by this cardnum in 30 days

actual/maximum amount by this cardnum in 30 days

Continued

actual/total amount by this cardnum in 30 days
actual/median amount by this cardnum in 30 days
actual-average amount by this merchnum in 30 days
actual amount by this merchnum in 30 days
actual/average amount by this merchnum in 30 days
actual/maximum amount by this merchnum in 30 days
actual/total amount by this merchnum in 30 days
actual/median amount by this merchnum in 30 days
actual-average amount by this cardnum at this merchnum in 30 days
actual/average amount by this cardnum at this merchnum in 30 days
actual-average amount by this cardnum in this zip code in 30 days
actual-average amount by this cardnum in this state in 30 days
actual amount by this cardnum in this zip code in 30 days
actual/average amount by this cardnum in this zip code in 30 days
actual/maximum amount by this cardnum in this zip code in 30 days
actual/total amount by this cardnum in this zip code in 30 days
actual/median amount by this cardnum in this zip code in 30 days
actual amount by this cardnum in this state in 30 days
actual/average amount by this cardnum in this state in 30 days
actual/maximum amount by this cardnum in this state in 30 days
actual/total amount by this cardnum in this state in 30 days
actual/median amount by this cardnum in this state in 30 days
actual amount by this cardnum at this merchnum in 30 days
actual/maximum amount by this cardnum at this merchnum in 30 days
actual/total amount by this cardnum at this merchnum in 30 days
actual/median amount by this cardnum at this merchnum in 30 days
total amount by this cardnum at this merchnum in 1 day
total amount by this cardnum at this merchnum in 7 days
total amount by this cardnum at this merchnum in 14 days
total amount by this cardnum in this zip code in 14 days
total amount by this cardnum in this zip code in 7 days
total amount by this cardnum in this zip code in 1 day
actual/total amount by this cardnum in this zip code in 1 day
actual/total amount by this cardnum in this zip code in 7 days
actual/total amount by this cardnum in this zip code in 14 days
actual/total amount by this cardnum at this merchnum in 14 days
actual/total amount by this cardnum at this merchnum in 7 days
actual/total amount by this cardnum at this merchnum in 1 day

Continued

actual/maximum amount by this cardnum at this merchnum in 1 day
actual/maximum amount by this cardnum at this merchnum in 7 days
actual/maximum amount by this cardnum at this merchnum in 14 days
actual/maximum amount by this cardnum in this zip code in 14 days
actual/maximum amount by this cardnum in this zip code in 7 days
actual/maximum amount by this cardnum in this zip code in 1 day
maximum amount by this cardnum in this zip code in 1 day
maximum amount by this cardnum in this zip code in 7 days
maximum amount by this cardnum in this zip code in 14 days
maximum amount by this cardnum at this merchnum in 14 days
maximum amount by this cardnum at this merchnum in 7 days
maximum amount by this cardnum at this merchnum in 1 day
average amount by this cardnum at this merchnum in 1 day
average amount by this cardnum at this merchnum in 7 days
average amount by this cardnum at this merchnum in 14 days
actual-average amount by this cardnum at this merchnum in 14 days
actual-average amount by this cardnum at this merchnum in 7 days
actual-average amount by this cardnum at this merchnum in 1 day
actual/average amount by this cardnum at this merchnum in 1 day
actual/average amount by this cardnum at this merchnum in 7 days
actual/average amount by this cardnum at this merchnum in 14 days
actual/average amount by this cardnum in this zip code in 14 days
actual/average amount by this cardnum in this zip code in 7 days
actual/average amount by this cardnum in this zip code in 1 day
actual-average amount by this cardnum in this zip code in 1 day
actual-average amount by this cardnum in this zip code in 7 days
actual-average amount by this cardnum in this zip code in 14 days
average amount by this cardnum in this zip code in 14 days
average amount by this cardnum in this zip code in 7 days
average amount by this cardnum in this zip code in 1 day
median amount by this cardnum in this zip code in 1 day
median amount by this cardnum in this zip code in 7 days
median amount by this cardnum at this merchnum in 14 days
median amount by this cardnum at this merchnum in 7 days
median amount by this cardnum at this merchnum in 1 day
median amount by this cardnum in this zip code in 14 days
actual-median amount by this cardnum in this zip code in 1 day
actual-median amount by this cardnum in this zip code in 7 days

Continued

actual-median amount by this cardnum in this zip code in 14 days
actual-median amount by this cardnum at this merchnum in 14 days
actual-median amount by this cardnum at this merchnum in 7 days
actual-median amount by this cardnum at this merchnum in 1 day
actual/median amount by this cardnum at this merchnum in 1 day
actual/median amount by this cardnum at this merchnum in 7 days
actual/median amount by this cardnum at this merchnum in 14 days
actual/median amount by this cardnum in this zip code in 14 days
actual/median amount by this cardnum in this zip code in 7 days
actual/median amount by this cardnum in this zip code in 1 day
Current transaction date - date of most recent transaction with same cardnum and state
Current transaction date - date of most recent transaction with same cardnum and zip
Current transaction date - date of most recent transaction with same cardnum and merchnum
Current transaction date - date of most recent transaction with same cardnum
Current transaction date - date of most recent transaction with same merchnum
average amount by this cardnum in this state in 1 day
average amount by this cardnum in this state in 7 days
average amount by this cardnum in this state in 14 days
maximum amount by this cardnum in this state in 1 day
maximum amount by this cardnum in this state in 7 days
maximum amount by this cardnum in this state in 14 days
total amount by this cardnum in this state in 1 day
total amount by this cardnum in this state in 7 days
total amount by this cardnum in this state in 14 days
median amount by this cardnum in this state in 1 day
median amount by this cardnum in this state in 7 days
median amount by this cardnum in this state in 14 days
actual-average amount by this cardnum in this state in 1 day
actual-average amount by this cardnum in this state in 7 days
actual-average amount by this cardnum in this state in 14 days
actual-median amount by this cardnum in this state in 1 day
actual-median amount by this cardnum in this state in 7 days
actual-median amount by this cardnum in this state in 14 days
actual/average amount by this cardnum in this state in 1 day
actual/average amount by this cardnum in this state in 7 days
actual/average amount by this cardnum in this state in 14 days
actual/maximum amount by this cardnum in this state in 1 day
actual/maximum amount by this cardnum in this state in 7 days

Continued

actual/maximum amount by this cardnum in this state in 14 days
actual/total amount by this cardnum in this state in 1 day
actual/total amount by this cardnum in this state in 7 days
actual/total amount by this cardnum in this state in 14 days
actual/median amount by this cardnum in this state in 1 day
actual/median amount by this cardnum in this state in 7 days
actual/median amount by this cardnum in this state in 14 days
average amount by this cardnum in 1 day
average amount by this cardnum in 7 days
average amount by this cardnum in 14 days
average amount by this merchnum in 1 day
average amount by this merchnum in 7 days
average amount by this merchnum in 14 days
maximum amount by this cardnum in 1 day
maximum amount by this cardnum in 7 days
maximum amount by this cardnum in 14 days
maximum amount by this merchnum in 1 day
maximum amount by this merchnum in 7 days
maximum amount by this merchnum in 14 days
median amount by this cardnum in 1 day
median amount by this cardnum in 7 days
median amount by this cardnum in 14 days
median amount by this merchnum in 1 day
median amount by this merchnum in 7 days
median amount by this merchnum in 14 days
total amount by this cardnum in 1 day
total amount by this cardnum in 7 days
total amount by this cardnum in 14 days
total amount by this merchnum in 1 day
total amount by this merchnum in 7 days
total amount by this merchnum in 14 days
actual-average amount by this cardnum in 1 day
actual-average amount by this cardnum in 7 days
actual-average amount by this cardnum in 14 days
actual-average amount by this merchnum in 1 day
actual-average amount by this merchnum in 7 days
actual-average amount by this merchnum in 14 days
actual-median amount by this cardnum in 1 day

Continued

actual-median amount by this cardnum in 7 days
actual-median amount by this cardnum in 14 days
actual-median amount by this merchnum in 1 day
actual-median amount by this merchnum in 7 days
actual-median amount by this merchnum in 14 days
actual/average amount by this cardnum in 1 day
actual/average amount by this cardnum in 7 days
actual/average amount by this cardnum in 14 days
actual/average amount by this merchnum in 1 day
actual/average amount by this merchnum in 7 days
actual/average amount by this merchnum in 14 days
actual/maximum amount by this cardnum in 1 day
actual/maximum amount by this cardnum in 7 days
actual/maximum amount by this cardnum in 14 days
actual/maximum amount by this merchnum in 1 day
actual/maximum amount by this merchnum in 7 days
actual/maximum amount by this merchnum in 14 days
actual/total amount by this cardnum in 1 day
actual/total amount by this cardnum in 7 days
actual/total amount by this cardnum in 14 days
actual/total amount by this merchnum in 1 day
actual/total amount by this merchnum in 7 days
actual/total amount by this merchnum in 14 days
actual/median amount by this cardnum in 1 day
actual/median amount by this cardnum in 7 days
actual/median amount by this cardnum in 14 days
actual/median amount by this merchnum in 1 day
actual/median amount by this merchnum in 7 days
actual/median amount by this merchnum in 14 days
#transactions with same cardnum in 1 day
#transactions with same cardnum in 7 days
#transactions with same cardnum in 14 days
#transactions with same merchnum in 1 day
#transactions with same merchnum in 7 days
#transactions with same merchnum in 14 days
#transactions with same cardnum at this merchnum in 1 day
#transactions with samebcardnum at this merchnum in 7 days
#transactions with same cardnum at this merchnum in 14 days

Continued

#transactions with same cardnum in this zip code in 1 day
#transactions with same cardnum in this zip code in 7 days
#transactions with same cardnum in this zip code in 14 days
#transactions with same cardnum in this state in 1 day
#transactions with same cardnum in this state in 7 days
#transactions with same cardnum in this state in 14 days