Deep Learning Detecting Fraud in Credit Card Transactions

Abhimanyu Roy, Jingyi Sun, Robert Mahoney, Loreto Alonzi, Stephen Adams, Peter Beling
University of Virginia, {ar3dd, js6sm, rlm4bj, lpa2a, sca2c, pb3a}@virginia.edu

Abstract **- Credit card fraud resulted in the loss of $3 billion to North American financial institutions in 2017. The rise of digital payments systems such as Apple Pay, Android Pay, and Venmo has meant that loss due to fraudulent activity is expected to increase. Deep Learning presents a promising solution to the problem of credit card fraud detection by enabling institutions to make optimal use of their historic customer data as well as real-time transaction details that are recorded at the time of the transaction. In 2017, a study found that a Deep Learning approach provided comparable results to prevailing fraud detection methods such as Gradient Boosted Trees and Logistic Regression. However, Deep Learning encompasses a number of topologies. Additionally, the various parameters used to construct the model (e.g. the number of neurons in the hidden layer of a neural network) also influence its results. In this paper, we evaluate a subsection of Deep Learning topologies – from the general artificial neural network to topologies with built-in time and memory components such as Long Short-term memory – and different parameters with regard to their efficacy in fraud detection on a dataset of nearly 80 million credit card transactions that have been pre-labeled as fraudulent and legitimate. We utilize a high performance, distributed cloud computing environment to navigate past common fraud detection problems such as class imbalance and scalability. Our analysis provides a comprehensive guide to sensitivity analysis of model parameters with regard to performance in fraud detection. We also present a framework for parameter tuning of Deep Learning topologies for credit card fraud detection to enable financial institutions to reduce losses by preventing fraudulent activity.**

Index Terms - Deep Learning, Fraud Detection, Neural Networks, Sensitivity Analysis

INTRODUCTION

This study aims to evaluate different Deep Learning topologies with regard to their efficacy in detecting credit card fraud. We analyze the various parameters that are used to construct the model to find the optimal combination of parameters to detect fraudulent activity. Deep Learning algorithms are a class of machine learning algorithms that use multiple non-linear processing units for feature extraction and transformation. These processing units discover intermediate representations in a hierarchical manner. The features discovered in one layer form the basis for processing of the succeeding layer. In this way, Deep Learning algorithms learn intermediate concepts between raw input and target [1].

Deep Learning initially found use in automatic speech recognition, image recognition and natural language processing [2]-[4]. More recently, Deep Learning algorithms have been used in areas that require the prediction of human behavior such as customer relationship management, recommendation systems and mobile advertising [5]-[7]. However, all these applications require choosing between different Deep Learning topologies (i.e., the structure of the model) as well as the parameters that are used to construct the model [8][9].

An area where Deep Learning can have a major impact is the detection of fraud in financial transactions. The Association for Financial Professionals' 2016 Payments Fraud and Control Survey found that 73 percent of finance professionals reported an attempted or actual payments fraud in 2015 [10]. This has been compounded further by the rise of digital payments systems such as Venmo, Apple Pay, Android Pay etc. A report by the Boston Consulting Group found that credit card fraud had cost North American financial institutions $3 billion in 2017 [11]. In April 2017, an article by consulting firm McKinsey concluded that Deep Learning presented a promising solution to the problem of financial fraud detection by enabling institutions to make optimal use of all of their historical customer data as well as real-time transaction details that are recorded at the time of the transaction [12].

This paper is organized in the following fashion. The following section provides a summary of related work. We then describe our research methodology and feature engineering. Next, we provide an overview of the different Deep Learning topologies we are investigating as well as the different parameter tuning options that we are evaluating. Finally, we present the results of our analysis and a discussion of the implications of our work and areas for further investigation.

RELATED WORKS

Anomaly detection in consumer behavior has been addressed previously using decision trees, Bayesian methods, clustering algorithms such as k-Nearest Neighbors, Neural Networks, Support Vector Machines, Regression models, Restricted Boltzmann Machines (RBMs) gradient boosted trees and Markov models. Recently, deep-learning algorithms such as Deep Belief Networks, Long Short-term Memory (LSTM) and recurrent neural networks have been shown to hold promise in this field. However, analysis and evaluation of the

978-1-5386-6343-1/18/$31.00 ©2018 IEEE

application of different deep-learning topologies in behavioral anomaly detection still remains largely unexplored.

Plötz, Hammerla and Olivier used RBMs for activity recognition in context-aware computing applications [13]. Lotfi *et al*. used recurrent neural networks to predict future values (start time and duration) of the activities and anomalous activity prediction of elderly patients and found very promising results [14]. Fang and Hu used RBMs to predict human activity in smart homes and found RBMs outperformed the Naive Bayes Classifier (NBC) in most of cases [15]. Arifoglu and Bouchachia used LSTM and Gated Recurrent Units (GRUs) for abnormal behavior detection among patients suffering from dementia and found they performed better than Hidden Markov Models (HMM) [16]. Lang and Rettenmeier investigated the applications of LSTM in eCommerce and recommendation systems and found that they performed better than NBC and HMM [17].

Some of the aforementioned studies such as the one by Lotfi *et al*. used data regarding physical human movement, which were obtained from sensors that were placed on the bodies of subjects [14]. While others such as the one by Lang and Rettenmeier used data regarding online behavior [17]. In this work, we evaluate a variety of Deep Learning topologies to determine their efficacy in credit card fraud detection.

Applications of predictive modelling in fraudulent activity detection has long been an important field of study. For instance, Chan *et al.* used a number of supervised learning approaches in credit card fraud detection [18]. Senator *et al.* used unsupervised clustering algorithms to detect money laundering [19]. Srivastava *et al*. used HMM in credit card fraud detection in eCommerce transactions and found that HMM detected fraud correctly 80% of the cases [20]. In 2011, Bhattacharya *et al*. used Support Vector Machine and Random Forest in detecting credit card detection and benchmarked the two methods against results obtained from Logistic Regression [21]. They found that Random Forest performed slightly better than the other two methods and produced more accurate results. In 1994, Ghosh and Reilly used a neural network to detect fraud in credit card transactions [22]. However, benchmarking of the performance of neural networks in fraud detection against other approaches remained largely unexplored until Patidar and Sharma found that artificial neural networks produced better results than Random Forest and other classification methods [23]. Rushin *et al*. used deep learning algorithms (Autoencoders) in fraud detection and found that the results produced were better than gradient boosted trees and logistic regression [24]. However, the implementation and evaluation of other deep learning topologies in fraud detection remains largely unexplored. In our study, we analyze a set of different deep learning topologies for fraud detection.

DATA

The data set used in this study was provided by a financial institution engaged in the retail banking. It contains details of nearly 80 million anonymized credit card transactions across an eight-month time period. The information provided includes both transaction and account details such as the transaction amount, the type of merchant the transaction was initiated from, the date the account was opened etc. The response variable has been pre-labeled as 1 for fraudulent transaction and 0 for non-fraudulent transaction. There is a significant class imbalance problem associated with our data set. Only 0.14% of the transactions are classified as fraudulent compared with 99.86% which are classified as legitimate.

I. Data Preprocessing

There was extensive missing data in the datetime and categorical variables. Transactions with missing data were not removed from the data set used in the analysis. For observations with missing categorical variables, the mode of the variable was used. Dummy variables were also added to indicate the missing data for categorical and time variables that had missing data.

II. Feature Engineering

The creation of domain expertise features has proven to improve the predictive accuracy of models in detecting credit card fraud [25]. Although banks collect an immense amount of data during the processing of credit card transactions, not all of this data can be directly translated into important predictors. There are many important features that can be derived from the original data set. In our research, we identified and created a few important and industry standard predictors.

The features created for this study expand upon the features created in [24]. The following predictors were created and added to the data:

- Frequency of transaction per month - these variables add information about an account's spending behavior
- Missing data dummy variables - missing data is an important predictor in credit card fraud detection because fraudsters suppress information on card applications during purchase
- Dummy variables to indicate when a purchase occurs at a specific merchant - certain merchants such as gas stations and restaurants are common places fraudsters test a stolen card before making a larger purchase
- Account history variables such as number of transactions from the account in the 8-month period covered in the dataset, maximum and mean authorization amount for the account.
- A dummy variable that indicate if a transaction authorization amount made at a given merchant is greater than 10% of the standard deviation of the mean of the non-fraudulent transactions for the merchant and the transaction deviation from the non-fraud mean is in the direction of the fraudulent mean. A 10% deviation was used because it has been found to be statistically significant in maximizing classification accuracy [26]-[29].

III. Undersampling

As stated previously, the dataset suffers from class imbalance. After creation of the new features, the legitimate transactions were undersampled at the account level. We thought there may be a temporal component in the data. Accordingly, we needed to make sure each account we sampled had all its transactions present in the training set. This involved separating the dataset into fraudulent and non-fraudulent datasets and extracting unique account numbers from the fraudulent dataset. Thereafter we randomly sampled non-fraudulent transactions from the non-fraudulent dataset for the extracted account numbers. The sampling ratio chosen was 10:1 (Non-fraudulent: fraudulent) which has been shown to be ideal for credit card fraud detection [30]. Finally, we used one-hot encoding to represent the categorical variables.

IV. Deep Learning Environment Specifications

The Deep learning environment consisted of 16 GPUs, 64 CPUs and 732GB of RAM. The operating system used was Ubuntu 16.04. To facilitate faster processing through the use of GPUs, we used Nvidia CUDA Toolkit 9.1 and Cuda Deep Neural Network (cuDNN) library 7.1. On top of this hardware setup, we installed an iPython server and used the Keras library, which uses the Tensorflow library as a backend to implement recurrent neural networks. Keras was used as it has a wrapper class than enables the use of Scikit-learn's grid searching capabilities for parameter tuning.

METHODS

Our analysis includes four different Deep Learning topologies: Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs), Long Short-term Memory (LSTMs), and Gated Recurrent Units (GRUs).

I. Artificial Neural Networks

In this paper, we utilize the Feed Forward Multilayer Perceptron neural network. This type of neural network consists of different layers of perceptrons (processing units) that are interconnected by a set of connections which have different weights associated with them. There are three types of layers in this network – an input layer that receives an input stream, the hidden layer which receives input from the input layer or from another hidden layer and an output layer, which produces the final output. There is full connectivity between the perceptrons of two consecutive layers. Signals can be propagated in two directions – forward from input layer to the hidden layer(s) to the output layer and error signals that are propagated backwards i.e., from the output layer, through the hidden layer(s) to the input layer. Important ANN parameters that affect the quality of the output are the number of neurons in the input, hidden and output layers, activation function, loss function and learning rate [31]. See figure I for a diagram of an ANN

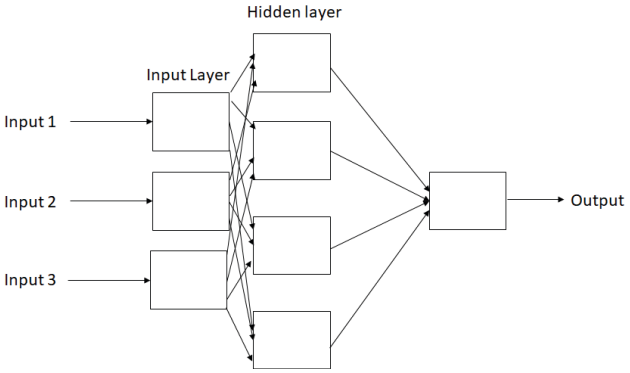

FIGURE I
ANN WITH 1 HIDDEN LAYER

II. Recurrent Neural Networks

Recurrent Neural Networks are a variant of Artificial Neural Networks that are adapted to the modelling of sequential data. Artificial neural networks do not offer the scalability required to model large sequential data [32]. In addition to links between layers, recurrent neural networks allow for the formation of links between neurons co-located in the same layer, which results in the formation of cycles in the network's architecture. Cycles allow the neurons in the model to share weights throughout successive values of a given input at different time steps. This allows for the activation function to take into account the state of the neuron at a previous stage in time. Thus, the state can be used to transfer some aspects of the previous time stages to upcoming time stages. Important parameters that affect the performance of RNNs are activation function, dropout rate and loss function [33]. See figure II for a diagram of a RNN

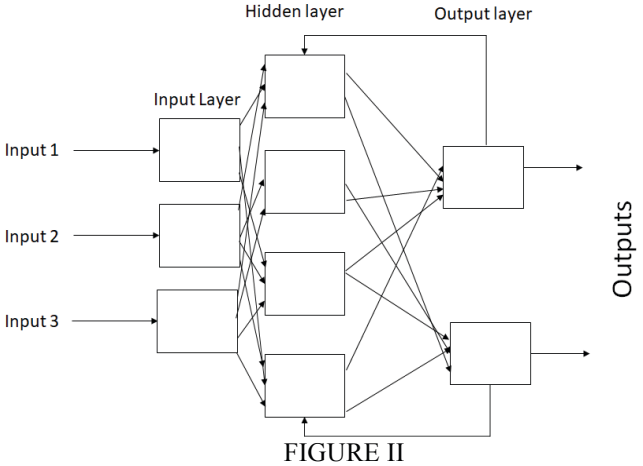

FIGURE II
RNN WITH 1 HIDDEN LAYER

III. Long Short-term Memory

Standard RNNs suffer from vanishing or exploding gradient problems. To address these issues, the Long Short-term Memory architecture was proposed. LSTMs contain a memory cell which maintains its state over time. Additionally, gating units are used to regulate the information flow into and out of the memory cell. More specifically, an input gate can allow the input signal to adjust the cell state or

prevent it (i.e., sets the input gate to zero). An output gate can allow the cell state to affect neurons in the hidden layers or block it. A forget gate enables the cell to remember or forget its previous state. Initially. the relative importance of each component is not clear [34]. Important LSTM parameters that affect the quality of the output are the number of neurons in the hidden layers, activation function and inner activation function and dropout rate [35]. See figure III for a diagram of an LSTM cell.

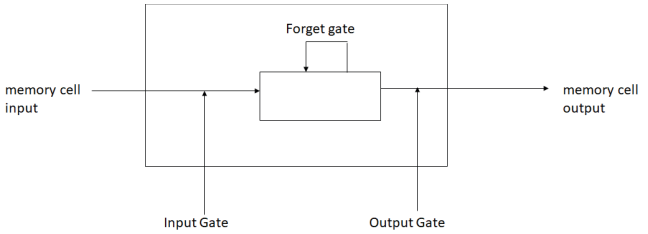

FIGURE III
LSTM CELL

IV. Gated Recurrent Unit

A gated recurrent unit makes each recurrent unit adaptively capture dependencies of different time scales. Although similar to an LSTM, the GRU has gating units that modulate information flow into the unit, however, GRUs do not have separate memory cells. Unlike LSTMs, GRUs expose their whole state each time [36]. The same parameters that affect LSTMs apply to GRUs too [37].

RESULTS

Overall the GRU architecture produced the single best performing model based on 10-fold cross validated accuracy score of 0.916. The model contained 6 hidden layers each with 150 nodes. This model was trained using a learning rate of 0.05, the *tanh* activation functions and mean absolute error as the error function. The best performing LSTM model contained 6 hidden layers each with 150 nodes each had an accuracy score of 0.912. This model was trained using a learning rate of 0.5, the tanh activation functions and mean absolute error as the error function. The best performing RNN model contained 6 hidden layers each with 150 nodes and had an accuracy score of 0.90433. This model was trained using a learning rate of 0.5, the *tanh* activation functions and mean absolute error as the error function. The ANN produced the worst results among the 4 topologies with an accuracy score of 0.889. The best performing ANN model contained 6 hidden layers each with 150 nodes. This model was trained using a learning rate of .05, an alpha of .001, the indemnity activation function, and lbfgs as the solver.

We also conducted sensitivity analysis to determine which hyperparameters had the largest impact on model performance. We found that size of the network to be the largest driver of model performance. Larger network tended to perform better than smaller networks. This is seen in each of 4 topologies having their best performing model resulting from using the largest network size in our search space. Box plots for the ANN and LSTM models are provided in figure IV and figure V. Although there is some variation in performance within each network size, the average performance for both topologies increase with an increase in network size.

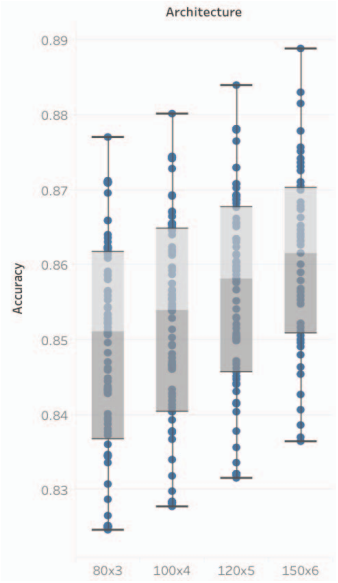

FIGURE IV
BOX PLOT OF ANN ACCURACY BY 240 400, 600, AND 900 NODES

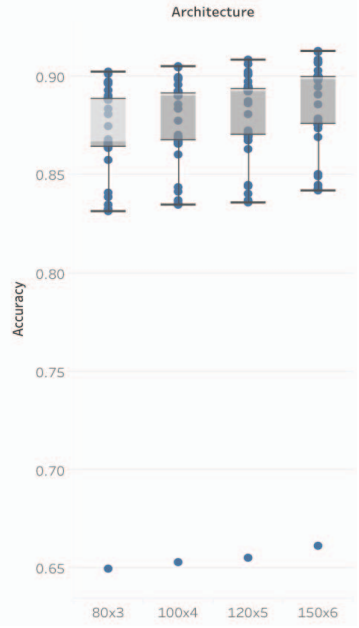

FIGURE V
BOX PLOT OF LSTM ACCURACY BY 240 400, 600, AND 900 NODES

The other hyperparameters had less impact on the model performance. The impact of the learning rate was inconsistent among the topologies. The best performing GRU and ANN model had a learning rate of .05. In contrast, the best result for the LSTM and RNN models were achieved with a learning rate of .5.

The loss functions performed comparably across the 4 topologies. Although their average performance was close, the binary cross entropy and the cosine proximity loss

functions led to a much wider range of model accuracy scores. For example, GRU models with trained with a binary cross entropy or cosine proximity loss function had a range of .084 and .077 respectively among accuracy scores. The hinge and mean absolute error loss functions had a comparably tighter range of accuracy scores at .02 and .033 respectively

CONCLUSIONS

Our analysis shows that there is a significant temporal component. The LSTM and GRU model significantly outperformed the baseline ANN which indicates that order of transactions for an account contains useful information in differentiating between fraud and non-fraudulent transactions. This suggests that that computational resources may be best spent training a larger network.

In our testing, performance improved whenever network size was increased. Further research is needed to determine when model performances cease to improve with network size increases. Further research could also be directed towards assessing model sensitivity to hyperparameters not included in our initial grid search such as momentum, batch size, number of epochs, and dropout rate. Our analysis held the number of neurons constant for each hidden layer. Varying the number of neurons in each layer may also reveal additional insight into the effect of network size on model performance.

ACKNOWLEDGMENT

We are grateful to the University of Virginia advisors who validated results. Additionally, we would like to thank Capital One for insight and guidance on the use case for this paper.

REFERENCES

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

[2] Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82-97.

[3] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[4] Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 160-167). ACM.

[5] Ngai, E. W., Xiu, L., & Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert systems with applications*, *36*(2), 2592-2602.

[6] Wang, H., Wang, N., & Yeung, D. Y. (2015, August). Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1235-1244). ACM.

[7] Zhai, S., Chang, K. H., Zhang, R., & Zhang, Z. M. (2016, August). Deepintent: Learning attentions for online advertising with recurrent neural networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1295-1304). ACM.

[8] Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems* (pp. 1135-1143).

[9] Denil, M., Shakibi, B., Dinh, L., & De Freitas, N. (2013). Predicting parameters in deep learning. In *Advances in neural information processing systems* (pp. 2148-2156).

[10] "Payments Fraud and Control Survey", *PNC Financial Services Group*, 2016. pnc.com/content/dam/pnc-com/pdf/corporateandinstitutional/Treasury%20Management/2016_AFP_Payments_Fraud_Report.pdf. Accessed: March 30, 2018.

[11] Boston Consulting Group. (2017). Global Payments 2017 – Deepening The Customer Relationship. Retrieved from image-src.bcg.com/Images/BCG-Global-Payments-2017-Oct-2017_tcm9-173047.pdf

[12] Corbo, J., Giovine, C., & Wigley, C. (April 2017). Applying analytics in financial institutions' fight against fraud. In McKinsey Analytics Retrieved February 8, 2018, from mckinsey.com/business-functions/mckinsey-analytics/our-insights/applying-analytics-in-financial-institutions-fight-against-fraud.

[13] Plötz, T., Hammerla, N. Y., & Olivier, P. (2011, July). Feature learning for activity recognition in ubiquitous computing. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence* (Vol. 22, No. 1, p. 1729).

[14] Lotfi, A., Langensiepen, C., Mahmoud, S. M., & Akhlaghinia, M. J. (2012). Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour. *Journal of ambient intelligence and humanized computing*, *3*(3), 205-218.

[15] Fang, H., & Hu, C. (2014, July). Recognizing human activity in smart home using deep learning algorithm. In *Control Conference (CCC), 2014 33rd Chinese* (pp. 4716-4720). IEEE.

[16] Arifoglu, D., & Bouchachia, A. (2017). Activity Recognition and Abnormal Behaviour Detection with Recurrent Neural Networks. *Procedia Computer Science*, *110*, 86-93.

[17] Lang, T., & Rettenmeier, M. (2017). Understanding Consumer Behavior with Recurrent Neural Networks. In *Proceedings of the 3rd Workshop on Machine Learning Methods for Recommender Systems*. mlrec. org/2017/papers/paper2. pdf.

[18] Chan, P. K., & Stolfo, S. J. (1998, August). Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. In *KDD* (Vol. 1998, pp. 164-168).

[19] Senator, T.E., Goldberg, H.G., Wooton, J., Cottin,i M.A., Khan, A.U., Klinger, C.D., Llamas, W.M., Marrone, M.P., Wong, R.W. (1995). Financial Crimes Enforcement Network AI System (FAIS) Identifying Potential Money Laundering from Reports of Large Cash Transactions. *AI magazine*, *16*(4), 21.

[20] Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. (2008). Credit card fraud detection using hidden Markov model. *IEEE Transactions on dependable and secure computing*, *5*(1), 37-48.

[21] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, *50*(3), 602-613.

[22] Ghosh, S., & Reilly, D. L. (1994, January). Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on* (Vol. 3, pp. 621-630). IEEE.

[23] Patidar, R., & Sharma, L. (2011). Credit card fraud detection using neural network. *International Journal of Soft Computing and Engineering (IJSCE)*, *1*(32-38).

[24] Rushin, G., Stancil, C., Sun, M., Adams, S., & Beling, P. (2017, April). Horse race analysis in credit card fraud—deep learning, logistic regression, and Gradient Boosted Tree. In *Systems and Information Engineering Design Symposium (SIEDS), 2017* (pp. 117-121). IEEE.

[25] Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, *51*, 134-142.

[26] Bolton, R. J., & Hand, D. J. (2001). Unsupervised profiling methods for fraud detection. *Credit Scoring and Credit Control VII*, 235-255.

[27] Şahin, Y. G., & Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines.

[28] Tedder, K., & Owner, P. (2009). Now You See It, Now You Don't: A Review of Fraud Costs and Trends. *First Data Corporation White Paper*.

[29] Allred, Christopher, Hite, Kathryn, Fonzone, Stephen, et al. "Modeling and Data Analysis in the Credit Card Industry:

Bankruptcy, Fraud, and Collections." *IEEE Systems and Information Design Symposium.*
[30] Chen, Xue-wen, and Michael Wasikowski. "Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems." *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008.
[31] Mohanraj, M., Jayaraj, S., & Muraleedharan, C. (2015). Applications of artificial neural networks for thermal analysis of heat exchangers–a review. *International Journal of Thermal Sciences*, *90*, 150-172.
[32] I. Goodfellow *et al.*, *Deep learning*, 2016, MIT Press.
[33] Bengio, Y., Boulanger-Lewandowski, N., & Pascanu, R. (2013, May). Advances in optimizing recurrent networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* (pp. 8624-8628). IEEE.
[34] Wu, Z., & King, S. (2016, March). Investigating gated recurrent networks for speech synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on* (pp. 5140-5144). IEEE.
[35] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, *28*(10), 2222-2232.
[36] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
[37] Wen, Y., Zhang, W., Luo, R., & Wang, J. (2016). Learning text representation using recurrent convolutional neural network with highway layers. *arXiv preprint arXiv:1606.06905*.

AUTHOR INFORMATION

Abhimanyu Roy, M.S. Student, Data Science Institute, University of Virginia.
Jingyi Sun, M.S. Student, Data Science Institute, University of Virginia.
Robert Mahoney, M.S. Student, Data Science Institute, University of Virginia.
Loreto Alonzi, Senior Data Scientist, Library and Data Science Institute, University of Virginia.
Stephen Adams, Senior Scientist, Systems and Information Engineering, University of Virginia.
Peter Beling, Professor and Interim Chair, Department of Systems and Information Engineering, University of Virginia.