Maxout Neural Network for Big Data Medical Fraud Detection

Gabriel Castaneda, Paul Morris, Taghi M. Khoshgoftaar
Florida Atlantic University

***Abstract* — Globally, health care losses due to fraud rise every year, and for this reason fraud detection is an active research area that, in the U.S. alone, can potentially save billions of dollars. We explore the performance of multiple maxout activation variants on the big data medical fraud detection task using neural networks. Maxout networks have gained great success in many computer vision tasks, but there is limited work on other classification tasks. Our experiments compare Rectified Linear Unit, Leaky Rectified Linear Unit, Scaled Exponential Linear Unit, and hyperbolic tangent to four maxout variants. We evaluate the effectiveness of the activation functions on four U.S. Centers for Medicare and Medicaid Services datasets. Throughout this paper, we found that maxout networks are considerably slower to train compared to traditional activation functions. We find that on average, across all datasets, Scaled Exponential Linear Unit's classification performance is better than any maxout activation, and reported the lowest training time.**

Keywords—maxout network, activation functions, big data, medical fraud detection, neural networks.

I. INTRODUCTION

An activation function in a neural network (NN) is a transfer function that transforms the net input of a neuron into an output signal. The output signal is then used as an input in the next layer in the stack. The activation function introduces nonlinearities to convolutional neural networks (CNNs) [1], which are desirable for multi-layer networks to detect nonlinear features. Commonly used activation functions include sigmoid, hyperbolic tangent (tanh) and Rectified Linear Unit (ReLU) [2]. There is a lack of consensus on how to select a good activation function for a NN, and a specific function may not be suitable for all applications. Since an activation function is generally applied to the outputs of all neurons, its computational complexity will contribute heavily to the overall execution time [3]. Most research works on the activation functions are focused on the complexity of the nonlinearity that an activation function can provide [4] or how fast it can be executed [5], but often neglect the impact on different classification tasks.

NNs have successfully utilized sigmoidal units, but sigmoidal activation functions suffer from gradient saturation. The major drawback of the sigmoid and the tanh functions are that saturation regions at both ends yield very small gradients. With the increase of the slope parameter in sigmoid and tanh functions, the saturation regions get larger. The rectifier function also saturates, when inputs are negative. These saturation regions cause gradient diffusion and block gradients from propagating to deeper layers [6]. For this reason, different activation functions have been proposed for NN and CNN training. Compared to traditional activation functions, like the logistic sigmoid units or hyperbolic tangent units, which are anti-symmetric, ReLU is one-sided. This property encourages the network to be sparse (i.e. the outputs of the hidden units are sparse), and thus more biologically plausible [7]. ReLU has become the most commonly used activation function for CNNs because it results in significant performance improvements in multiple domains.

The maxout unit [8] selects the maximum value within a group of different feature maps and is usually combined with dropout [9], which is widely used to regularize deep networks to prevent overfitting. This technique randomly drops units or connections to prevent units from co-adapting. Dropout has been shown to improve classification accuracy in various computer vision tasks [10]. Maxout chooses the maximum of n copies of each feature in a network. The simplest case of maxout is the Max-Feature-Map (MFM) [11], where $n=2$. In the past five years, variants of maxout have been tested on benchmark datasets. In image classification, the Maxout network In Network (MIN) [12] showed that the MIN method achieves state-of-the-art or comparable performance on the Mixed National Institute of Standards and Technology (MNIST), the Canadian Institute for Advanced Research (CIFAR-10), CIFAR-100, and Street View House Numbers (SVHN) datasets. Maxout layers were applied in sentiment analysis [13], with a hybrid architecture consisting of a recurrent NN stacked on top of a CNN. This approach outperforms a standard convolutional deep neural architecture as well as a recurrent network architecture and performs competitively compared to other methods on two datasets of annotated customer reviews. Maxout is effective in image classification and sentiment analysis tasks. It could also benefit other domains such as medical fraud detection.

Medical fraud is often linked to insurance fraud, prescription fraud, the submission of claims for patients who are dead or who do not exist, and upcoding, where a doctor performs a medical procedure but changes the insurer for one that is more expensive, or perhaps does not even perform one at all [14]. Automatic fraud detection helps to reduce the manual parts of a fraud screening process, becoming one of the most established industry/government data mining applications [15]. In the United States, the Centers for Medicare and Medicaid Services (CMS) [16] released a few datasets for

978-1-7281-0059-3/19/$31.00 ©2019 IEEE
DOI 10.1109/BigDataService.2019.00064

different parts of the Medicare program for the detection of fraudulent activities. The Federal Bureau of Investigation (FBI) estimates that fraud accounts for 3–10% of healthcare costs [17]. In 2017, the Medicare spending was 15% of total federal spending with a total possible cost recovery (with the potential application of effective fraud detection methods) of $21 to $70 billion from Medicare alone [18]. NNs help to classify the practice profiles of practitioners who participate in medical practice to help identify those who are practicing inappropriately.

Most of the comparisons between maxout and other activation functions only report a single performance metric, ignore network size, and only report accuracy on a single dataset, with no training time or memory use analysis. Furthermore, it is unclear whether marginal performance gains with maxout are due to the activation function or simply an increase (2x) in the number of convolutional filters versus ReLU networks. In this work, we evaluated multiple activation functions for NNs applied to medical fraud detection. To the best of our knowledge, this is the first study to evaluate multiple maxout variants and standard activations for big data medical fraud detection with significance testing. The main contributions herein can be summarized as follows:

- Evaluate four maxout functions and compare them to popular activation functions like tanh, ReLU, LReLU, and SeLU.
- Compare training time for various activation functions.
- Evaluate whether marginal performance gains with maxout are due to the activation function or simply an increase (2x) in the number of convolutional filters versus ReLU networks.
- Determine whether maxout methods tend to converge faster and if they significantly outperform standard activation functions in terms of accuracy.

The remainder of this paper is organized as follows. Section II describes related work on activation function evaluation on multiple classification domains. Section III describes the evaluated activation functions. The datasets employed in our experiments are described in Section IV. The experimental methodology is presented in Section V. Results and analysis are provided in Section VI. Conclusions with some directions for future work are provided in Section VII.

II. Related Work

Most prior work focuses on proposing new activation functions, but few studies have compared different activation functions for big data medical fraud detection. Also, there are few comparisons between maxout and traditional activation functions. Most of the comparisons do not report the details of their network to indicate whether an increased number of filters was accounted for in the experiment and only report accuracy on a single or a couple of datasets.

He et al. [19], implemented a NN to detect the inappropriate practice of physicians within the Medicare program in Australia. The data consisted of 1,500 general practice (GP) profiles with 28 features. The profiles were randomly divided into two datasets with 750 profiles for the training set and 750 profiles for the test set. On the test set, the positive class had 290 instances and the negative class had 460. The NN only used a single hidden layer and two output classifications. The accuracy rate was 88.4% on the training set and 80.9% on the test set. Ortega et al. [20], worked with a private health insurance company in Chile to implement a fraud detection system that identifies 15 different types of fraud. A committee of 10 multilayer feedforward NNs was implemented, and the output of several independently trained networks was averaged to reduce their model variance. The architecture was 12-3-1 and all activation functions were sigmoidal. The dataset contains 418 fraudulent and 8,401 normal cases. The model was able to identify 73.4% of the true abuses and only 6.9% false positives.

Using the Public Use File (PUF) data from CMS, Branting et al. [21] proposed graph analysis as a framework for healthcare fraud risk assessment. Their algorithm was evaluated on the Part B (2012-2014), Part D (2013) and List of Excluded Individuals/Entities (LEIE) datasets. Using 10-fold cross-validation on the full 12,000-member and 11-feature dataset, they obtained a mean f-measure score of 0.919 and the mean AUC score of 0.960. Sadiq et al. [22] use the 2014 CMS Part B, Part D, and DMEPOS datasets (using only the provider claims from Florida) in order to find anomalies that possibly point to fraudulent or anomalous behavior. A novel framework based on Patient Rule Induction Method (PRIM) was presented, where abnormal behaviors of the physicians are detected. The experimental results show that their framework can effectively shrink the target dataset and identify a potential suspect subset of physicians who submit several anomalous claims and probably qualify as fraudsters. The attribute sub-space and their correlations are used in PRIM to characterize the low conditional probability region. The attribute space was characterized by PRIM, which provides a deeper understanding of how certain attributes are the key predictors in identifying fraud. Herland et al. [23] focused on the detection of Medicare fraud using the CMS Part B, Part D, and DMEPOS datasets. A fourth dataset was created by combining the three primary datasets. Based on the area under the ROC curve performance metric, their results show that the combined dataset with the Logistic Regression (LR) learner yielded the best overall score at 0.816, closely followed by the Part B dataset with LR at 0.805.

III. Activation Functions

In this section, we introduce each evaluated activation function used in our study. Subsections A through C provide a definition for Hyperbolic Tangent, Rectified Units and Maxout Units.

A. Hyperbolic Tangent

A hyperbolic tangent (tanh) function is a ratio between hyperbolic sine and cosine functions of x:

$$f(x) = \tanh = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (1)$$

B. Rectified Units

Rectified Linear Unit (ReLU) [2] is defined as:

$$h(x_i) = \max(0, x_i) \quad (2)$$

where x_i is the input and $h(x_i)$ is the output. The ReLU activation is the identity for positive arguments and zero otherwise.

Leaky ReLU (LReLU) [24] assigns a slope to its negative input. It is defined as:

$$h(x_i) = \min(0, a_i x_i) + \max(0, x_i) \quad (3)$$

where $a_i \in (0, 1)$ is a predefined slope.

The scaled exponential linear unit (SeLU) [25] is given by:

$$SeLU(x) = \lambda \begin{cases} x & if\ X > 0 \\ \alpha e^x - \alpha & if\ X \leq 0 \end{cases} \quad (4)$$

where x is used to indicate the input to the activation function. Klambauer et al. [25] justify why α and λ must have the below values:

$$\begin{aligned} \propto &= 1.6732632423543772848170429916717 \\ \lambda &= 1.0507009873554804934193349852946 \end{aligned} \quad (5)$$

to ensure that the neuron activations converge automatically toward an average of 0 and a variance of 1.

C. Maxout Units

A maxout unit takes as input the output of multiple linear functions and returns the largest:

$$h(x_i) = \max_{k \in \{1, \ldots, k\}} w^k \cdot x_i + b^k \quad (6)$$

In theory, maxout can approximate any convex function [8], but a large number of extra parameters introduced by the k linear functions of each hidden maxout unit result in large storage memory cost and considerable training time, which affect the training efficiency of very deep CNNs. For our comparisons we use four variants of the maxout activation: an activation with k = 2 input neurons for every output (maxout 2-1), an activation with k = 3 input neurons for every output (maxout 3-1), an activation with k = 6 input neurons for every output (maxout 6-1), and a variant of maxout with k = 3 where the two maximum neurons are selected (maxout 3-2). These maxout variants have proven to be effective in classification tasks such as image classification [26], facial recognition [11], and speech recognition [10].

IV. Datasets

In this section, we describe the datasets used in our study.

A. Medicare Part B

The CMS [16] released the Part B dataset [27] and describes Medicare provider claims information for the entire U.S. and its commonwealths, where each instance in the data shows the claims for a provider and procedure performed for a given year. Physicians are identified using their unique National Provider Number (NPI) [28], while procedures are labeled by their Healthcare Common Procedure Coding System (HCPCS) code [29]. Other claims information includes average payments and charges, the number of procedures performed and medical specialty (also known as provider type).

B. Medicare Part D

The Part D PUF [30] provides information on prescription drugs prescribed by individual physicians and other health care providers and paid for under the Medical Part D Prescription Drug Program. Each physician is denoted by his or her NPI and each drug is labeled by its brand and generic name. Other information includes average payments and charges, variables describing the drug quantity prescribed and medical specialty.

C. DMEPOS

The Durable Medical Equipment, Prosthetics, Orthotics and Supplies (DMEPOS) PUF [31] presents information on DMEPOS products and services provided to Medicare beneficiaries ordered by physicians and other healthcare professionals. Physicians are identified using their unique NPI within the data while products are labeled by their HCPCS code. Other claims information includes average payments and charges, the number of services/products rented or sold and medical specialty (also known as provider type).

D. Combined CMS dataset

A combined dataset was created in [23] after processing Part B, Part D, and the DMEPOS datasets, containing all the attributes from each, along with the fraud labels derived from the List of Excluded Individuals and Entities (LEIE). The combining process involves a join operation on NPI, provider type, and year. Due to there not being a gender variable present in the Part D data, the authors did not include this variable in the join operation condition and used the gender labels from Part B while removing the gender labels gathered from the DMEPOS dataset after joining. In combining these datasets, it is limited to those physicians who have participated in all three parts of Medicare.

E. Data Processing

For each dataset (Part B, Part D and DMEPOS), the information was combined for all available calendar years. For Part B and DMEPOS, all attributes not present in each available year were removed. The Part D dataset had the same attributes in all available years. For Part B, the standard deviation variables were removed from 2012 and 2013 and standardized payment variables were removed from 2014 and 2015 as they were not available in the other years. For DMEPOS, the standard deviation variable was removed from 2014 and 2015 as it was not available in 2013. For all three datasets, all instances that either were missing both NPI and HCPCS/drug name values or had an invalid NPI were removed. For Part B, all instances with HCPCS codes referring to prescriptions were filtered out. The prescription-related codes are not actual medical procedures, but instead are for specific services listed on the Medicare Part B Drug Average Sales Price file. For the Part B dataset, eight features were kept while the other twenty-two were removed. For the Part D dataset, seven features were kept and the other fourteen were removed. For the DMEPOS dataset nine features were kept and the other nineteen were removed. The excluded attributes provide no specific information on the claims, drugs administered, or referrals, but rather encompass provider-related information, such as location and name, as well as redundant variables like text descriptions which can be represented by using the variables containing the procedure or drug codes. For Part D, variables that provided count and payment information for patients 65 or older were not included, as this information is encompassed in the retained variables. The combined dataset contains all the retained features from all three datasets. The purpose of this new dataset is to provide a more encompassing view into a physician's behavior over various branches of Medicare, over individual Medicare parts.

V. Empirical Methodology

We evaluate classification performance with three fully-connected layers. Between layers, dropout is used to prevent overfitting. The NN architecture is presented in Table 1.

Layer	Medicare Part B Medicare Part D DMEPOS Combined CMS Dataset
Input	123
Fully-Connected	n=512
Drop Out	kp=0.5
Fully-Connected	n=64
Drop Out	kp=0.5
Fully-Connected	n=2

Table 1: NN Configuration. Dropout layers show the applied keep probability (kp=), and the fully-connected layers display the number of neurons (n=).

The reported results are generated with the models trained using a learning rate of 0.01. Rather than tune each network in our comparison optimally with a validation set, we implement a set of uniform stopping criteria during training to maintain a consistent protocol so that network performance on a test set is suitable for comparison across activations [32]. Early stopping criteria is the same for every dataset, with the slope of the test loss calculated over a running window of the past three epochs. When the slope goes positive, testing loss no longer decreases, and network training is stopped. The optimizer is stochastic gradient descent and the loss function is the categorical cross-entropy. The batch size for all datasets is 200 and the number of trainable parameters is 95,872.

ReLU is also evaluated with 2x the number of filters in each convolutional layer. The purpose of including this variant is to consider the impact of increased neurons on the accuracy, training time and memory usage of NN independent of the maxout activation. Because maxout incorporates both the max operation and the use of duplicate neurons with additional memory, it is necessary to consider how each component of the activation contributes to its performance.

Maxout is evaluated with the following input feature map-output elements combinations: 2-1, 3-1, 3-2 and 6-1. We compute maxout for our four activations using the equations below, which are suitable for parallelization with modern deep learning software and parallel computer hardware. In general, we use maximum (max) and minimum (min) operations with two inputs to achieve maximum computational efficiency during training.

$$maxout\ 2-1\ (x_1, x_2) = \max(x_1, x_2) \tag{7}$$

$$maxout\ 3-1\ (x_1, x_2, x_3) = \max(x_1, \max(x_2, x_3)) \tag{8}$$

$$maxout\ 6-1\ (x_1, x_2, x_3, x_4 x_5, x_6) = \max(x_1, \max(x_2, \max(x_3, \max(x_4, \max(x_5, x_6))))) \tag{9}$$

$$maxout\ 3-2\ (x_1, x_2, x_3) = \max(x_1, \max(x_2, x_3)), \min(\max(x_1, x_2), \min(\max(x_2, x_3), \max(x_1, x_3))) \tag{10}$$

While it would be ideal to record the wall clock time needed to train each network, modern high-performance computing environments present hardware and software challenges which make it difficult to safely compare training time across runs or activations. Thus, we produce a metric which represents the time cost of training with a particular activation function. This metric is produced for each activation on an isolated desktop computing environment. We record the wall clock time required to train each network in our comparison for 100 batches and take the average time across 10 runs on a single desktop computer with 32GB of RAM running Ubuntu 16.04 with an intel i7 7th generation CPU and an NVIDIA 1080ti GPU. Those times are produced independently for each activation on each dataset.

A total of nine activation functions were evaluated where each experiment compared:

- Classification accuracy
- Training time (100 batches time multiplied by number of epochs to converge)

There are five experiments per activation function and dataset. In each dataset, we use a train/test split of 90:10. Because we apply a consistent early stopping criterion, we report results of our comparison done directly on a test set, without an additional validation set. We implemented our tests in PyTorch [33].

VI. Experimental Results

As the Medicare datasets are highly imbalanced, we employ random undersampling (RUS) to mitigate the adverse effects of class imbalance. RUS is the process of randomly removing instances from the majority class of a dataset in order to balance the ratio (non-fraudulent/fraudulent). We generate a class distribution (majority:minority) of 50:50. There are 2036 samples in Medicare Part D, 2818 in Medicare part B, 1275 in DMEPOS and 946 in the CMS combined dataset.

One-way analysis of variance (ANOVA) [34] is performed to statistically examine the various effects on performances of the type of activation (maxout vs. other activations) across all the datasets. In this ANOVA test, the results from 180 evaluations were considered together, and all tests of statistical significance utilized a significance level α of 5%. The factor is significant if the *p*-value is less than 0.05. The ANOVA table is presented in Table 2, indicating the activation type does not make a difference in the classification accuracy.

Factors	Sum of Squares	Percentage of Variation	Degrees of Freedom	Mean Square	F-Computed	*p*-value
Activation Type	0.15	0.01 %	1	0.1534	0.02	0.87
Error	1103.37	99.98 %	178	6.19873		
Total	1103.53	100 %	179			

Table 2: One-way ANOVA for type of activation and classification task

The best activation accuracy per dataset, its average to train 100 batches and training time are presented in Table 3. The training time is number of epochs to converge multiplied by the average time in seconds needed to train 100 batches. SeLU reported the highest accuracy on Medicare Part B, DMEPOS and the combined CMS datasets. On Medicare Part D, maxout 2-1 and maxout 6-1 achieved the highest accuracy. On average, SeLU reported the highest accuracy of 69.7%. This suggests that SeLU is effective for the medical fraud detection task using a NN.

The difference between the worst and best activations are 7.5% and 4.5% on the Medicare Part B and Medicare Part D datasets respectively. On the combined and DMEPOS datasets, the difference was similar with DMEPOS (4.5%) and the combined CMS dataset (6%). On the Medicare Part D dataset, maxout 2-1 and maxout 6-1 obtained the highest accuracy, followed by SeLU, ReLU2x and maxout 3-2 with a 0.5% difference in value. This confirms that SeLU is also effective in this dataset.

Dataset	Highest Activation	Average Accuracy	Average Epochs	Average 100 Batches Time (s)	Average 100 Batches Training Time (s)
Medicare Part B	SeLU	71.0 %	29	0.12	7.98
Medicare Part D	Maxout 2-1 Maxout 6-1	71.5 %	180	0.12	22.84
DMEPOS	SeLU	68.5 %	51	0.12	5.54
Combined CMS dataset	SeLU	74.0 %	160	0.12	21.05
All datasets combined	SeLU	69.7 %	107	0.12	13.01

Table 3: Best activation accuracy, 100 batches average and training time per dataset

We performed Tukey's Honestly Significantly Difference (HSD) test to further investigate these results. The HSD is a statistical test comparing the mean value of the performance measure for the different activation functions. All tests of statistical significance use an $\alpha = 5\%$. Two activation functions with the same block letter are not significantly different with 95% statistical confidence (e.g. group a is significantly different than group b). In Table 4, the letters in the third column indicate the HSD grouping of the activation accuracy. That is, if two activations have the same letter in the HSD column, their accuracies are not significantly different. The HSD test shows eight activations are statistically indistinguishable from one another (they all have the block letter 'b' in the HSD column). All the maxout and ReLU variants had a similar performance, and SeLU performed better than maxout 6-1, ReLU and LReLU.

Activation	Average Accuracy	Accuracy HSD	Average 100 Batches Time (s)	Average 100 Batches Training Time (s)
SeLU	69.70 %	a	0.12	13.01
Tanh	69.02 %	ab	0.11	22.62
Maxout 3-1	68.97 %	ab	0.14	24.77
Maxout 2-1	68.55 %	ab	0.12	21.50
ReLU 2x	68.55 %	ab	0.11	24.66
Maxout 3-2	68.52 %	ab	0.19	30.79
Maxout 6-1	68.37 %	b	0.13	25.05
ReLU	68.22 %	b	0.11	23.53
LReLU	67.82 %	b	0.12	27.15

Table 4: HSD test, 100 batches average and training time on medical datasets

VII. Conclusion

We conducted experiments to assess the effectiveness of maxout variants (using operations other than max or using more weights). Our experiments suggest that given the medical fraud detection task on a NN architecture, SeLU is likely to produce the best classification accuracy results compared to the rest of the activation functions analyzed in this study. It is important to note that the difference between ReLU and ReLU2x is a choice of a tunable hyperparameter. In this study, due to hardware memory constraints, we exclude ReLU3x and ReLU6x from comparisons.

SeLU reported the fastest average 100 batches training time. On average, SeLU tends to converge 1.7x faster than tanh, and

2.3x faster than maxout 3-2, which reported the slowest training time of all activations. There is no relationship between the activation functions that use more memory or have a higher training time and the classification accuracy performance. SeLU is the recommended activation function due to high performance, and fast training relative to other top performing activations.

Future work will involve conducting additional empirical studies with ReLU3x and ReLU6x on big data and hyperparameter tuning recommendations that were outside the scope of this work. Also, future work could include additional deep network architectures and domains.

REFERENCES

[1] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," The handbook of brain theory and neural networks, Cambridge, MA USA, 1995.

[2] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010.

[3] S. Liew, M. Khalil-Hani and R. Bakhteri, "Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems," *Neurocomputing,* vol. 216, pp. 718-734, 2016.

[4] S. Sodhi and P. Chandra, "Bi-modal derivative activation function for sigmoidal feedforward networks," *Neurocomputing,* vol. 143, pp. 182-196, 2014.

[5] V. Nambiar, M. Khalil-Hani, R. Sahnoun and M. Marsono, "Hardware implementation of evolvable block-based neural networks utilizing a cost efficient sigmoid-like activation function," *Neurocomputing,* vol. 140, pp. 228-241, 2014.

[6] J. Li, W. Ng, D. Yeung and P. Chan, " Bi-firing deep neural networks," *International Journal of Machine Learning and Cybernetics,* vol. 5, no. 1, pp. 73-83, 2014.

[7] Y. Li, P. Ding and B. Li, "Training Neural Networks by Using Power Linear Units (PoLUs)," *arXiv preprint arXiv:1802.00212.,* 2018.

[8] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville and Y. Bengio, "Maxout Networks," in *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, 2013.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[10] M. Cai, Y. Shi and J. Liu, "Deep maxout neural networks for speech recognition," in *Proc. ASRU*, 2013.

[11] X. Wu, R. He, Z. Sun and T. Tan, "A light CNN for deep face representation with noisy labels," arXiv preprint arXiv:1511.02683, 2015.

[12] J. Chang and Y. Chen, "Batch-normalized maxout network in network," arXiv preprint arXiv:1511.02583, 2015.

[13] S. Jebbara and P. Cimiano, "Aspect-Based Relational Sentiment Analysis Using a Stacked Neural Network Architecture," arXiv preprint arXiv:1709.06309, 2017.

[14] R. Bolton and D. Hand, "Statistical fraud detection: A review," *Statistical science,* vol. 17, no. 3, pp. 235-249, 2002.

[15] C. Phua, V. Lee, K. Smith and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," in *arXiv preprint arXiv:1009.6119*, 2010.

[16] Centers for Medicare and Medicaid Services, "Center for medicare and medicaid services," [Online]. Available: https://www.cms.gov/. [Accessed 2018].

[17] L. Morris, "Combating fraud in health care: an essential component of any cost containment strategy," *Health Affairs,* vol. 28, no. 5, pp. 1351-1356, 2009.

[18] Henry J Kaiser Family Foundation, "The Facts on Medicare Spending and Financing," [Online]. Available: https://www.kff.org/medicare/issue-brief/the-facts-on-medicare-spending-and-financing/. [Accessed 10 12 2018].

[19] H. He, J. Wang, W. Graco and S. Hawkins, "Application of neural networks to detection of medical fraud," *Expert Systems with Applications,* vol. 13, no. 4, pp. 329-336, 1997.

[20] P. Ortega, C. Figueroa and G. Ruz, "A Medical Claim Fraud/Abuse Detection System based on Data Mining: A Case Study in Chile," in *Conference on Data Mining (DMIN)*, 2006.

[21] L. Branting, F. Reeder, J. Gold and T. Champney, "Graph analytics for healthcare fraud risk estimation," in *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2016.

[22] S. Sadiq, Y. Tao, Y. Yan and M. Shyu, "Mining Anomalies in Medicare Big Data Using Patient Rule Induction Method," in *Multimedia Big Data (BigMM), 2017 IEEE Third International Conference on*, 2017.

[23] M. Herland, T. M. Khoshgoftaar and R. Bauder, "Big Data fraud detection using multiple medicare data sources," *Journal of Big Data,* vol. 5, no. 1, 2018.

[24] A. Maas, A. Hannun and A. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013.

[25] G. Klambauer, T. Unterthiner, A. Mayr and S. Hochreiter, "Self-normalizing neural networks," *Advances in Neural Information Processing Systems,* pp. 971-980, 2017.

[26] I. Goodfellow, M. Mirza, D. Xiao, A. Courville and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211,* 2013.

[27] Centers for Medicare and Medicaid Services, "Medicare Provider Utilization and Payment Data: Physician and Other Supplier," [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Physician-and-Other-Supplier.html. [Accessed June 2018].

[28] CMS, "National Provider Identifier Standard," [Online]. Available: https://www.cms.gov/Regulations-and-Guidance/Administrative-Simplification/NationalProvIdentStand/. [Accessed 4 11 2018].

[29] CMS, "HCPCS - General Information," [Online]. Available: https://www.cms.gov/Medicare/Coding/MedHCPCSGenInfo/index.html. [Accessed 4 11 2018].

[30] Centers for Medicare and Medicaid Services, "Medicare Provider Utilization and Payment Data: Part D Prescriber," [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Part-D-Prescriber.html. [Accessed June 2018].

[31] CMS, "Medicare Provider Utilization and Payment Data: Referring Durable Medical Equipment, Prosthetics, Orthotics and Supplies," [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/DME.html. [Accessed 4 11 2018].

[32] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Technical report, University of Toronto.,* vol. 1, no. 4, p. 7, 2009.

[33] R. Collobert, K. Kavukcuoglu and C. Farabet, "Torch7: A matlab-like environment for machine learning," Advances in Neural nformation Processing Systems, 2011.

[34] M. L. Berenson, D. M. Levine and M. Goldstein, Intermediate Statistical Methods and Applications: A Computer Package Approach, Prentice-Hall, Inc., 1983.