

Artificial Neural Network Technique for Improving Prediction of Credit Card Default: A Stacked Sparse Autoencoder Approach

S. A. Ebiaredoh-Mienye¹, E. Esenogho^{*2}, Theo G. Swart³,

^{1, 2, 3} Centre for Telecommunication, Dept. of Electrical and Electronic Engineering Science, University of Johannesburg
P. O. Box 524, Auckland Park, 2006, South Africa

Article Info

Article history:

Received

Revised

Accepted

Keywords:

Sparse Autoencoder

Artificial Neural Network

Deep learning

Machine Learning

Unsupervised learning

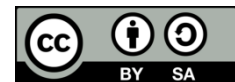
Feature learning

Credit card default

ABSTRACT

Presently, credit card the use has become a critical part of contemporary banking and predicting potential credit card defaulters or debtors is a crucial business opportunity for financial institutions. Meanwhile, some machine learning methods have been applied to achieve this task. However, with the dynamic and imbalanced nature of credit card default statistics, it is challenging for classical machine learning algorithms to proffer robust models with optimal performance. Research has shown that the performance of machine learning algorithms can be significantly improved when provided with optimal features. In this paper, we propose an unsupervised feature learning method to improve the performance of various classifiers using a stacked sparse autoencoder (SSAE) approach. The SSAE was optimized to achieve improved performance. The proposed SSAE learned accurate feature representations which were used to train the classifiers. The performance of the proposed approach is compared with an instance where the classifiers were trained using the raw data. Also, a comparison is made with previous scholarly works, and the proposed approach showed superior performance over other methods.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author: Ebenezer Esenegho,

Centre for Telecommunication, Department of Electrical and Electronic Engineering Science,
University of Johannesburg, Kingsway Avenue and University Road, Auckland Park,
Johannesburg 2006, South Africa.

Email: ebenezer@uj.ac.za

1. INTRODUCTION

In artificial intelligence and machine learning, tasks such as classification and clustering, the input data tends to influence the performance of the algorithms. Optimal performance is obtained when algorithms are given suitable data. To this end, some machine learning methods focus on processing high dimensional data, including linear dimensionality reduction methods such as linear discriminant analysis, principal component analysis, and multiple dimensional scaling and nonlinear dimensionality reduction methods such as isometric feature mapping, locally linear embedding, and Laplacian Eigenmap. Meanwhile, feature engineering and representation learning are the two main methods used to achieve representation from raw data. Recent research has focused on the latter since feature engineering methods are usually dependent on domain knowledge, are labor-intensive, and time-consuming [1]. Furthermore, representation learning methods tend to learn a representation from data automatically, which can then be used for classification. An autoencoder (AE) is a type of unsupervised representation learning.

Autoencoders are unsupervised neural networks with three layers, including an input layer, a hidden layer, and an output layer. The output layer can be considered as the reconstruction layer [2]. The structure of a basic autoencoder is shown in Figure 1. Autoencoders tend to learn a representation of the input data, usually for dimensionality reduction, by training the network to ignore noise. Together with the reduction side, a reconstructing side is learned, where the AE attempts to create a representation that is as close as possible to the original input [3]. There are different types of autoencoders, including sparse, denoising, contractive, variational, and convolutional autoencoders [4].

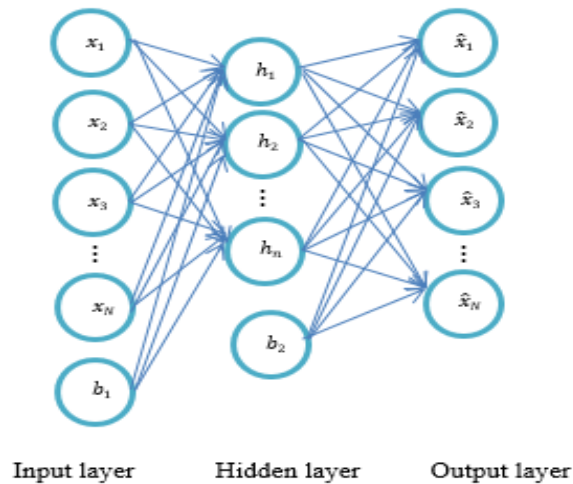


Figure 1. Structure of an Autoencoder

Credit card default/fraud detection is a crucial problem that has gotten the attention of machine learning researchers, and a significant number of approaches have been proposed [5]–[7]. However, the problem is still challenging since most credit card data seem to suffer from class imbalance as non-fraud transactions overwhelmingly supersede fraud transactions, making it difficult for many machine learning algorithms to achieve good performance. Meanwhile, a good feature representation can be obtained from the dataset, which can enhance the classification performance of the algorithms. Representation learning is a possible solution to the challenge of credit card default and fraud prediction because of its remarkable feature learning ability in large and unbalanced datasets.

While basic autoencoders aim at learning a representation or encode data by training the network to ignore noise and reconstructing the data as close as possible to the input data, however, training the autoencoder network in such a way that encourages sparsity can result in optimal feature learning. Sparsity induced neural networks have been extensively applied in image recognition and several other applications resulting in state-of-the-art performance [8]–[10].

In this paper, we propose an approach to improve the classification performance of classifiers by using the unsupervised feature learning capability of autoencoders. During the training of the autoencoder, sparsity is encouraged, and the model is optimized using the AdaMax algorithm [11] instead of the conventional stochastic gradient descent. To ensure accurate feature representation, we stack two sparse autoencoders to get the final model. Also, to further prevent overfitting and enhance the performance, speed, and stability of the network, we introduced the batch normalization technique [12] to the network. The low-dimensional features are then used to train various classifiers, including logistic regression (LR), classification and regression tree (CART), k -nearest neighbor (KNN), support vector machine (SVM), and linear discriminant analysis (LDA). The performance of the proposed method is compared with an instance where the classifiers were trained with the raw data. Further comparison is made with other scholarly works, and our proposed method shows better performance.

2. RELATED WORKS

Recently, autoencoders have been applied to several tasks, and they achieved state-of-the-art performance. In this section, we discuss some previous works that utilized various autoencoders and lay the foundation for the proposed stacked sparse autoencoder network. Sun et al. [13] proposed a method for fault diagnosis by applying a sparse stacked denoising autoencoder for feature extraction due to its robustness and data reconstruction capability, which improved the diagnostic accuracy. The autoencoder was used together with an optimized transfer learning algorithm. Similarly, Zhu et al. [14] proposed a novel stacked pruning sparse denoising autoencoder for intelligent fault diagnosis of rolling bearings. The method comprised of a fully connected autoencoder network, connecting the optimal features extracted from previous layers to subsequent layers. To effectively train the autoencoder, a pruning operation was added to the model to restrict

non-superior units from participating in all subsequent layers. When compared with other fault diagnostic models, their approach showed superior performance.

Furthermore, Sankaran et al. [15] proposed a feature extraction method using an autoencoder network, and $\ell_{2,1}$ -norm based regularization was used to achieve sparsity. The authors identified that due to the presence of many training parameters, several feature learning models are susceptible to overfitting, and different regularization approaches have been studied in literature to mitigate overfitting in deep learning models. The performance of their model was studied on publicly available latent fingerprint datasets, and it gave an improved performance. Chen et al. [16] proposed a method to address the challenge of learning efficiency and computational complexity in deep neural networks. The technique used a deep sparse autoencoder network to learn facial features and softmax regression applied to classify expression features. The softmax regression aimed at handling extensive data in the output of the autoencoder network. Also, to overcome local extrema and the challenge of gradient diffusion during training, the network weights were fine-tuned to achieve global optimum, and this improved the performance of the architecture.

Most approaches used to implement autoencoders depend on the single autoencoder model, and this presents a problem when learning different characteristics of data. Yang et al. [17] proposed a method to solve the problem by implementing a feature learning framework using serial autoencoders. The technique achieved superior representation learning by serially connecting two different types of autoencoders. The approach incorporated two encoding stages using a marginalized denoising autoencoder and a stacked robust autoencoder via graph regularization. When compared to baseline methods, the proposed approach showed significant improvement. Meanwhile, Al-Hmouz et al. [18], introduced a logic-driven autoencoder, whereby the network structure was achieved using some fuzzy logic operations. The autoencoder was also optimized using gradient-based learning. Lastly, sparse autoencoder networks have achieved remarkable performance in representation learning [19]–[22]. However, better representation learning can be gotten when multiple sparse autoencoders are stacked and optimized effectively, which is the focus of this research.

3. PROPOSED METHODOLOGY

This section considers the methodology applied to developing the proposed autoencoder. An autoencoder consists of two functions, i.e., an encoder and decoder, the former maps the d -dimensional input data to get a hidden representation, and the latter maps the hidden representation back to a d -dimensional vector that is as close as possible to the encoder input [23]. Assuming the original input is x , the autoencoder encodes it into a hidden layer h to reduce the input dimension, which is then decoded at the output. The input vector is encoded according to:

$$h = \sigma(Wx + b) \quad (1)$$

where σ represents the activation function; in this case, the sigmoid activation function, W is the weight matrix, and b is a bias vector. The hidden representation is decoded to get the data as close as possible to the input x using:

$$\hat{x} = \sigma(W'h + b') \quad (2)$$

where W' is weight matrix and b' represents the bias vector. The sigmoid activation function is described as:

$$\sigma = \frac{1}{1 + e^{-x}} \quad (3)$$

The disparity between the original input x and the reconstructed input \hat{x} is called the reconstruction error. To optimize the parameters W , W' , b , b' , the mean squared error (MSE) function is used as the reconstruction error function:

$$E = \frac{1}{N} \sum_{i=1}^N \|\hat{x}_i - x_i\|^2 \quad (4)$$

The problem of overfitting associated with autoencoder networks is solved by applying a weight penalty to the cost function as:

$$E = \frac{1}{N} \sum_{i=1}^N \frac{1}{2} \|\hat{x}_i - x_i\|^2 + \frac{\lambda}{2} (||W||^2 + ||W'||^2) \quad (5)$$

where λ is the coefficient for L2 regularization, and it prevents overfitting. The average activation of neurons in the hidden layer is represented as:

$$\hat{\rho}_j = \frac{1}{N} \sum_{i=1}^N h_j(x_i) \quad (6)$$

To induce sparsity, we limit $\hat{\rho}_j = \rho$, where ρ is the sparsity proportion, and it is usually a small positive number near 0. Therefore, we try to minimize the Kullback–Leibler (KL) divergence between $\hat{\rho}_j$ and ρ according to:

$$KL(\hat{\rho}||\rho) = \sum_{j=1}^S \rho \log\left(\frac{\rho}{\hat{\rho}_j}\right) + (1 - \rho) \log\left(\frac{1 - \rho}{1 - \hat{\rho}_j}\right) \quad (7)$$

where S represents the total number of neurons in a layer. Hence, the overall cost function of the sparse autoencoder becomes:

$$E = \frac{1}{N} \sum_{i=1}^N \frac{1}{2} \|\hat{x}_i + x_i\|^2 + \frac{\lambda}{2} (||W||^2 + ||W'||^2) + \beta KL(\hat{\rho}||\rho) \quad (8)$$

where β is the sparsity regularization parameter, and it sets the sparsity penalty term. We now have three optimization parameters, including β , λ , and ρ , and we set their values as 3, 0.0001, and 0.05, respectively. In the sparse autoencoder network, a neuron is said to be active if its output is a value close to 1, while it is inactive if its output is a value closer to 0 [8]. Algorithm 1 shows the proposed sparse autoencoder procedure.

Algorithm 1. Proposed methodology for sparse autoencoder

Input:

train set x

Process:

1. Start
2. Initialize σ, W, W', b, b'
3. Obtain the cost function according to (4)
4. Apply weight penalty to the cost function according to (5)
5. Add the sparsity regularizer to the cost function according to (7)
6. Train network until convergence
7. End

Output:

Reconstructed representation of the input

The AdaMax algorithm [11], a variant of the adaptive moment estimation (Adam) algorithm that uses the infinity norm, was applied to optimize the sparse autoencoder. Also, we introduced a batch normalization technique [12] to prevent overfitting further and enhance the performance, speed, and stability of the network. Lastly, to further improve the feature representation and, by extension, enhance the performance of the classifiers, we stack two sparse autoencoders. A stacked sparse autoencoder can comprise of multiple sparse autoencoders whereby the outputs of each layer are connected to the inputs of the next layer [24]. Figure 2 shows the structure of the proposed stacked sparse autoencoder (SSAE). For simplicity, the decoder parts of the SAE are not shown. The output of the SSAE is then used to train the various classifiers.

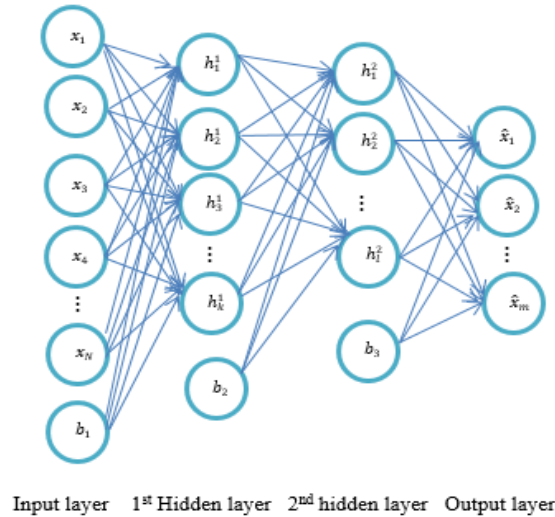


Figure 2. Structure of the proposed SSAE model

4. CASE STUDY OF CREDIT CARD DEFAULTING PREDICTION MODELS

Credit risk plays a crucial role in the financial industry. Most financial institutions grant loans, mortgage, and credit cards, among many other services. Due to the rising number of credit card clients, these institutions have faced an increasing default rate. They are thereby resorting to the use of machine learning methods to automate the application process and predict the probability of a client's future default. However, several machine learning methods have been developed in various literature with varying performance. A major limitation to achieving optimal performance in the credit card default prediction is that the datasets are highly imbalanced, i.e., the instances where clients do not default supersede the default cases.

Certain studies have used the default of credit card clients dataset [25] and achieved good performance. For example, Prusti and Rath [26] used various algorithms such as decision tree, k -nearest neighbor, support vector machine, and multilayer perceptron to make predictions on the dataset. Additionally, they proposed a method that hybridized decision tree, support vector machine, and k -nearest neighbor, which gave improved performance compared to the stand-alone algorithms. Sayjadah et al. [27] conducted a performance evaluation of credit card default prediction using logistic regression, random forest, and decision tree. The experimental results showed that random forest achieved superior performance with an accuracy of about 82%.

Furthermore, because the dataset is imbalanced, a method is proposed to tackle the problem using synthetic minority over-sampling technique (SMOTE) [28]. Using the SMOTE method together with seven other algorithms, the random forest algorithm achieved the best performance with an accuracy of 89.01% and F1-score of 89%. Lastly, Hsu et al. [29] and Chishti and Awan [30] also proposed models to predict the defaulting of credit card clients and achieved comparable performance. However, we are aiming to improve on what has been done by applying our proposed method on the same dataset.

5. RESULTS AND DISCUSSIONS

In this work, the defaulting of the credit card client dataset [25] is used. The dataset was obtained from the University of California Irvine (UCI) machine learning repository, and it contains 30,000 instances and 25 attributes, including demographic and financial records. The dataset was established to predict customers who are likely to default on payments in Taiwan. Out of the 30,000 instances 23,364 are non-default and 6,636 are default cases. The rationale behind the dataset is for financial institutions to be able to identify possible customers who will default on their credit card payments, thereby declining such applications. We use the 70-30% train-test split. The SSAE is trained with the training set in an unsupervised fashion, while the test set is input with the learned SSAE model to obtain the low-dimensional data. The classifiers are then trained using the low-dimensional train set, and the performance tested using the low-dimensional test set. The number of neurons in the first and second hidden layers was set at 100 and 85, respectively.

To efficiently evaluate the performance of our approach, we utilize performance metrics such as accuracy, sensitivity, precision, and F1 score. Accuracy is the ratio of the number of correct predictions to the total number of predictions made, sensitivity is the ratio of the number of correct positive predictions to the total actual positives, precision is the number of correct positive predictions divided by the number of positive results

predicted, and **F1** score is the harmonic mean between precision and sensitivity. Mathematically, the performance metrics can be represented as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (10)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (11)$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \quad (12)$$

Therefore, it can be represented as

$$\text{F1 score} = \left[\frac{2 \left[\left(\frac{TP}{TP + FP} \right) \left(\frac{TP}{TP + FN} \right) \right]}{\left(\frac{TP}{TP + FP} \right) + \left(\frac{TP}{TP + FN} \right)} \right] \quad (13)$$

For simplicity, **F1 score** in Equation (13) can be derived or represented as

$$\text{F1 score} = \left[\frac{2TP}{2TP + (FP + FN)} \right] \quad (14)$$

See appendix for the complete derivation of **F1 score**.

where **TP**, **TN**, **FP**, **FN** stands for the number of true positives, the number of true negatives, the number of false positives, and the number of false negatives, respectively. Meanwhile, all the experiments were carried out using a computer with the following specifications: Intel Core i5-6300U, 2.40 GHz, with 16 GB RAM. And Python programming language was used for the computations. To show the effectiveness of the proposed approach, we conduct a comparative study with five base classifiers. Therefore, we first show the performance of these classifiers on the raw dataset. The classifiers include CART, LR, KNN, SVM, and LDA, and the results are shown in Table 1.

Table 1. Performance of the base classifiers on the dataset

Algorithm	Accuracy (%)	Precision (%)	Sensitivity (%)	F1 score (%)
LR	78	62	78	69
CART	73	74	73	73
KNN	75	71	75	72
SVM	36	74	36	35
LDA	81	79	81	78

Table 2 summarizes the results obtained when the classifiers are trained using the features learned from the stacked sparse autoencoder. It can be seen that the learned features significantly improves the performance of the classifiers. Furthermore, the results show the ability of the proposed SSAE to learn a good representation of the data.

Table 2. Impact of the features learned by the SSAE on the base classifiers

Algorithm	Accuracy (%)	Precision (%)	Sensitivity (%)	F1 score (%)
LR	90	87	91	89
CART	86	84	84	84
KNN	89	87	90	88
SVM	88	86	88	87
LDA	90	91	90	90

To further show the effectiveness of the proposed method, the best performing model from our experiments, which is the LDA, is used to compare with other well-performing methods proposed in recent studies that have been discussed in Section 4. To give a fair comparison, we focused on studies that used similar datasets. This comparison is shown in Table 3, and it can be seen that our method outperforms those in the stated literature.

Table 3. Comparison with other methods

Literature	Accuracy (%)	Precision (%)	Sensitivity (%)	F1 score (%)
Prusti and Rath [26]	82.58	96.83	83.57	89.71
Sayjadah et al. [27]	81.81	-	-	-
Subasi and Cankurt [28]	89.01	-	-	89
Hsu et al. [29]	80.2	-	-	-
Chishti and Awan [30]	82	84	96	89
Proposed SSAE+LDA	90	91	90	90

From the above results, we can see that our proposed approach achieved better performance compared to the other methods. The improved performance can be attributed to the proposed SSAE that was able to learn a good representation of the original input data. Also, the results have shown the capability of deep learning in achieving exceptional performance in different tasks, including feature representation. Lastly, this study has demonstrated the importance of training machine learning algorithms with suitable data, and that improved performance can be obtained not only by hyper-parameter tuning but also and more efficiently by effective feature learning.

6. CONCLUSION

Conventional machine learning algorithms are often ineffective in performing classification on large datasets such as most credit card datasets. Hence, in this paper, a stacked sparse autoencoder is proposed to achieve optimal feature learning. In the proposed autoencoder network, we introduced a batch normalization technique to enhance the performance, speed, and stability of the model and to prevent overfitting further. Also, the model was optimized using the AdaMax algorithm. The learned data was then used to train five shallow machine learning algorithms, and the performance tested. When compared with a case where the algorithms were trained with the raw data, our proposed method showed superior performance. Furthermore, the results were compared with methods in some literature that used a similar dataset, and the proposed approach also showed significant improvement. Future research will focus on studying the effect of different optimizers and stacking numerous sparse autoencoders and the resultant impact on the feature learning process.

ACKNOWLEDGEMENTS

This work is supported partially by the Center of Telecommunications, University of Johannesburg, South Africa.

AUTHOR'S-CONTRIBUTIONS

Conceptualization, S. A. E.; methodology, S. A. E., E. E, T. G, S.; software, S. A. E., E. E.; validation, S. A. E., E. E.; formal analysis, S. A. E., E. E, T. G, S.; investigation, S. A. E., E. E, T. G, S.; resources, E. E, T. G, S.; data curation, E. E, T. G, S.; original draft preparation, S. A. E., E. E; review and editing, E. E, T. G, S.; visualization, E.E.; supervision, E. E, T. G, S.; project administration, E. E, T. G, S.; funding acquisition, E. E, T. G, S.,

FUNDING


This research received no external funding. The APC will be paid from our research center funds.

REFERENCES

- [1] A. S. Shamsabadi, M. Babaie-Zadeh, S. Z. Seyyedsalehi, H. R. Rabiee, and C. Jutten, "A new algorithm for training sparse autoencoders," in *2017 25th European Signal Processing Conference (EUSIPCO)*, Aug. 2017, pp. 2141–2145, doi: 10.23919/EUSIPCO.2017.8081588.
- [2] B. Yan and G. Han, "Effective Feature Extraction via Stacked Sparse Autoencoder to Improve Intrusion Detection System," *IEEE Access*, vol. 6, pp. 41238–41248, 2018, doi: 10.1109/ACCESS.2018.2858277.
- [3] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Proceedings of the 21th international conference on Artificial neural networks - Volume Part I*, Espoo, Finland, Jun. 2011, pp. 44–51, Accessed: Apr. 13, 2020. [Online].
- [4] G. Dong, G. Liao, H. Liu, and G. Kuang, "A Review of the Autoencoder and Its Variants: A Comparative Perspective from Target Recognition in Synthetic-Aperture Radar Images," *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 3, pp. 44–68, Sep. 2018, doi: 10.1109/MGRS.2018.2853555.
- [5] "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018, doi: 10.1109/TNNLS.2017.2736643.
- [6] C. Jiang, J. Song, G. Liu, L. Zheng, and W. Luan, "Credit Card Fraud Detection: A Novel Approach Using Aggregation Strategy and Feedback Mechanism," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3637–3647, Oct. 2018, doi: 10.1109/JIOT.2018.2816007.
- [7] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," *IEEE Access*, vol. 8, pp. 25579–25587, 2020, doi: 10.1109/ACCESS.2020.2971354.
- [8] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang, "Robust Single Image Super-Resolution via Deep Networks With Sparse Prior," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3194–3207, Jul. 2016, doi: 10.1109/TIP.2016.2564643.
- [9] M. Gong, J. Liu, H. Li, Q. Cai, and L. Su, "A Multiobjective Sparse Feature Learning Model for Deep Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3263–3277, Dec. 2015, doi: 10.1109/TNNLS.2015.2469673.
- [10] J. A. Ellis and S. Rajamanickam, "Scalable Inference for Sparse Deep Neural Networks using Kokkos Kernels," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep. 2019, pp. 1–7, doi: 10.1109/HPEC.2019.8916378.
- [11] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017, Accessed: Oct. 31, 2019. [Online]. Available: <http://arxiv.org/abs/1412.6980>.
- [12] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv:1502.03167 [cs]*, Mar. 2015, Accessed: Jan. 14, 2020. [Online]. Available: <http://arxiv.org/abs/1502.03167>.
- [13] M. Sun, H. Wang, P. Liu, S. Huang, and P. Fan, "A sparse stacked denoising autoencoder with optimized transfer learning applied to the fault diagnosis of rolling bearings," *Measurement*, vol. 146, pp. 305–314, Nov. 2019, doi: 10.1016/j.measurement.2019.06.029.
- [14] H. Zhu, J. Cheng, C. Zhang, J. Wu, and X. Shao, "Stacked pruning sparse denoising autoencoder based intelligent fault diagnosis of rolling bearings," *Applied Soft Computing*, vol. 88, p. 106060, Mar. 2020, doi: 10.1016/j.asoc.2019.106060.
- [15] A. Sankaran, M. Vatsa, R. Singh, and A. Majumdar, "Group sparse autoencoder," *Image and Vision Computing*, vol. 60, pp. 64–74, Apr. 2017, doi: 10.1016/j.imavis.2017.01.005.
- [16] L. Chen, M. Zhou, W. Su, M. Wu, J. She, and K. Hirota, "Softmax regression based deep sparse autoencoder network for facial emotion recognition in human-robot interaction," *Information Sciences*, vol. 428, pp. 49–61, Feb. 2018, doi: 10.1016/j.ins.2017.10.044.
- [17] S. Yang, Y. Zhang, Y. Zhu, P. Li, and X. Hu, "Representation learning via serial autoencoders for domain adaptation," *Neurocomputing*, vol. 351, pp. 1–9, Jul. 2019, doi: 10.1016/j.neucom.2019.03.056.
- [18] R. Al-Hmouz, W. Pedrycz, A. Balamash, and A. Morfeq, "Logic-driven autoencoders," *Knowledge-Based Systems*, vol. 183, p. 104874, Nov. 2019, doi: 10.1016/j.knosys.2019.104874.
- [19] S. Feng, H. Yu, and M. F. Duarte, "Autoencoder based sample selection for self-taught learning," *Knowledge-Based Systems*, vol. 192, p. 105343, Mar. 2020, doi: 10.1016/j.knosys.2019.105343.
- [20] B. Xu, H. Lin, Y. Lin, and K. Xu, "Incorporating query constraints for autoencoder enhanced ranking," *Neurocomputing*, vol. 356, pp. 142–150, Sep. 2019, doi: 10.1016/j.neucom.2019.03.068.
- [21] S. Liu *et al.*, "Discovering influential factors in variational autoencoders," *Pattern Recognition*, vol. 100, p. 107166, Apr. 2020, doi: 10.1016/j.patcog.2019.107166.
- [22] L. E. Escajeda Ochoa, I. B. Ruiz Quinde, J. P. Chuya Sumba, A. V. Guevara, and R. Morales-Menendez, "New Approach based on Autoencoders to Monitor the Tool Wear Condition" *IFAC-PapersOnLine*, vol. 52, no. 11, pp. 206–211, Jan. 2019, doi: 10.1016/j.ifacol.2019.09.142.
- [23] O. Irsoy and E. Alpaydm, "Unsupervised feature extraction with autoencoder trees," *Neurocomputing*, vol. 258, pp. 63–73, Oct. 2017, doi: 10.1016/j.neucom.2017.02.075.
- [24] J. Xu *et al.*, "Stacked Sparse Autoencoder (SSAE) for Nuclei Detection on Breast Cancer Histopathology Images," *IEEE Trans Med Imaging*, vol. 35, no. 1, pp. 119–130, Jan. 2016, doi: 10.1109/TMI.2015.2458702.
- [25] "UCI Machine Learning Repository: default of credit card clients Data Set."

- <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients> (accessed Mar. 14, 2020).
- [26] D. Prusti and S. K. Rath, "Web service based credit card fraud detection by applying machine learning techniques," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, Oct. 2019, pp. 492–497, doi: 10.1109/TENCON.2019.8929372.
- [27] Y. Sayjadah, I. A. T. Hashem, F. Alotaibi, and K. A. Kasmiran, "Credit Card Default Prediction using Machine Learning Techniques," in *2018 Fourth International Conference on Advances in Computing, Communication Automation (ICACCA)*, Oct. 2018, pp. 1–4, doi: 10.1109/ICACCAF.2018.8776802.
- [28] A. Subasi and S. Cankurt, "Prediction of default payment of credit card clients using Data Mining Techniques," in *2019 International Engineering Conference (IEC)*, Jun. 2019, pp. 115–120, doi: 10.1109/IEC47844.2019.8950597.
- [29] T.-C. Hsu, S.-T. Liou, Y.-P. Wang, Y.-S. Huang, and Che-Lin, "Enhanced Recurrent Neural Network for Combining Static and Dynamic Features for Credit Card Default Prediction," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 1572–1576, doi: 10.1109/ICASSP.2019.8682212.
- [30] W. A. Chishti and S. M. Awan, "Deep Neural Network a Step by Step Approach to Classify Credit Card Default Customer," in *2019 International Conference on Innovative Computing (ICIC)*, Nov. 2019, pp. 1–8, doi: 10.1109/ICIC48496.2019.8966723.

BIOGRAPHIES OF AUTHORS

	<p>S. A. Ebiaredoh-Mienye received her BS degree in Software Engineering in 2012 from the University of East London, UK. Currently, she is rounding up her Masters in Electrical and Electronic Engineering at the University of Johannesburg, South Africa. She is a researcher and teacher with over 5 years of experience. Her research interests include machine learning, deep learning, IoT, financial modeling and analytics.</p>
	<p>Dr. Ebenezer Esenogho received the Diploma in Computer Engineering (Upper Division) in 2003, B.Eng. degree (2nd Class Hons Upper Div.) in Computer Engineering in 2008, the M.Eng. degree (Upper Div.) in Electronic/Telecommunication Engineering from the University of Benin in 2012. He previously lectured at the University of Benin from March 2011 to 2013 and rose to rank of Lecturer I, before embarking on his Ph.D. program in 2013. Prior to now, he was involved in research and teaching with the Centre for Radio Access and Rural Technology, University of KwaZulu-Natal, Centre of Excellence, where he rounded-up his Ph.D. degree in Electronic (5G Cognitive Network) from 2013-2017 funded by Alcatel, Ericsson and Huawei. Currently, he is hired into the prestigious Global Excellence and Stature (GES) Post-Doctoral Research Fellowship at the University of Johannesburg, Auckland Park under the Institute for Intelligent System (IIS), Center for Telecommunication Research (CfT) to pursue further research. He is a recipient of several grants/scholarship/award/fellowships including the CEPS/Eskom's HVDC 2013, CEPS/Eskom's HVDC 2014, J. W Nelson 2015 and GES 2017, 2018, 2019/20. He is passionate about service to humanity and served in several community services both in the Republic of South Africa and his home country. His leadership service is evidence as a first postgraduate representative for Student Engineering Council in the Faculty management board of University of KwaZulu-Natal during his PhD days, 2016-2017. Presently, he serving as the first postdoctoral research fellowship representative in the University of Johannesburg Senate/Council, 2018-date. He was a UJ/DST/NRF research delegate to the H2020-ESASTAP EU-South Africa STI Cooperation on Strengthening Technology, Research and Innovation in Vienna, Austria. Dr. Ebenezer has authored/co-authored several peer-reviewed journals and conference papers, chaired sessions in conferences and, reviewed for some notable ISI/Scopus journals in his field. He is very familiar with ISO-9001/18001/14001/22301/27001/20000-1 with a strong affinity for project linked to R&D. His research interests are in but not limited to the Fifth Generation (5G) Wireless Networks, Cognitive Radio Networks, Network Security/Cybersecurity, Smart Grid Networks, IoT/IoE, SDN/SDR, Wireless Sensor Networks, Artificial Intelligence/Machine Learning, Big Data & Mobile/Cloud Computing and Visible light communication. He is a registered Engineer in Nigeria and a member of South Africa Institute of Electrical Engineers (SAIEE) and IEEE region 8.</p>



Prof. Theo G. Swart was born and educated in South Africa. He received the B. Ing. and M. Ing. degrees in electric and electronic engineering from the Rand Afrikaans University, South Africa, in 1999 and 2001, respectively (both cum laude), and the D. Ing. degree from the University of Johannesburg, South Africa, in 2006. He is an associate professor in the Department of Electrical and Electronic Engineering Science and the Director of the UJ Center for Telecommunications. He is a senior member of the IEEE and was previously the chair of the IEEE South Africa Chapter on Information Theory. He is also a specialist editor in Communications and Signal Processing for the SAIEE Africa Research Journal. To date he has over 20 journal articles and over 50 conference articles to his name, and has been the co-editor of and contributor to two editions of a comprehensive book on power-line communication. He has served as co-chair for the African Winter School on Information Theory and Communications in 2015 and 2019, technical programme co-chair for the IEEE International Symposium on Power Line Communications and its Applications in 2013, Technical programme track co-chair for IEEE Africon in 2017 and has been a TPC member of various other international conferences. He has done reviews for various journals and conferences, including reviews for the IEEE Transactions on Information Theory, IEEE Transactions on Communications, IEEE International Symposium on Information Theory, IEEE International Symposium on Power Line Communications and its Applications, IEEE Africon and SATNAC. His research interests include information theory, error correction coding, line coding, digital communications and power-line communications.

APPENDIX

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (10)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (11)$$

$$F1 \text{ score} = \frac{2 \times \text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \quad (12)$$

Therefore, it can be represented as

$$F1 \text{ score} = 2 \left[\frac{\left(\frac{TP}{TP + FP} \right) \left(\frac{TP}{TP + FN} \right)}{\left(\frac{TP}{TP + FP} \right) + \left(\frac{TP}{TP + FN} \right)} \right] \quad (13)$$

For simplicity, **F1 score** in Equation (13) can be derived as follows

$$2 \left[\frac{\left(\frac{TP}{TP + FP} \right) \left(\frac{TP}{TP + FN} \right)}{\left(\frac{TP}{TP + FP} \right) + \left(\frac{TP}{TP + FN} \right)} \right] = \frac{\frac{2(TP)^2}{(TP + FP)(TP + FN)}}{\left[\frac{2(TP)^2}{(TP + FP)(TP + FN)} \right] \div \left[\left(\frac{TP}{TP + FP} \right) + \left(\frac{TP}{TP + FN} \right) \right]}$$

Finding the lowest common factor

$$\left[\frac{2(TP)^2}{(TP + FP)(TP + FN)} \right] \div \left[\frac{TP(TP + FN) + TP(TP + FP)}{(TP + FP)(TP + FN)} \right]$$

Open the bracket

$$\left[\frac{2(TP)^2}{(TP + FP)(TP + FN)} \right] \div \left[\frac{(TP)^2 + TPFN + (TP)^2 + TPFP}{(TP + FP)(TP + FN)} \right]$$

Bring like terms together and factorize.

$$\frac{2(TP)^2}{(TP + FP)(TP + FN)} \div \frac{2(TP)^2 + TP(FN + FP)}{(TP + FP)(TP + FN)}$$

Invert and cancel out like terms

$$\frac{2(TP)^2}{(TP + FP)(TP + FN)} \times \frac{(TP + FP)(TP + FN)}{2(TP)^2 + TP(FN + FP)}$$

$$\frac{2(TP)^2}{2(TP)^2 + TP(FN + FP)}$$

Factorized TP out of the denominator

$$\frac{2(TP)^2}{TP[(2TP) + (FN + FP)]}$$

For simplicity we have,

$$F1 \text{ score} = \left[\frac{2TP}{2TP + (FP + FN)} \right] \quad (14)$$