Circulation in Computer Science
Vol.3, No.1, pp: (10-21), January 2018
https://doi.org/10.22632/ccs-2017-252-71

Financial Fraud Detection using Radial Basis Network

I. O. Alabi
Department of Information & Media Technology
Federal University of Technology
Minna, Nigeria

R. G. Jimoh
Department of Computer Science
University of Ilorin
Ilorin, Nigeria

ABSTRACT
The ubiquitous cases of abnormal transactions with intent to defraud is a global phenomenon. An architecture that enhances fraud detection using a radial basis function network was designed using a supervised data mining technique—radial basis function (RBF) network, a multivariate interpolation approximation method. Several base models were thus created, and in turn used in aggregation to select the optimum model using the misclassification error rate (MER), accuracy, sensitivity, specificity and receiver operating characteristics (ROC) metrics. The results shows that the model has a zero-tolerance for fraud with better prediction especially in cases where there were no fraud incidents doubtful cases were rather flagged than to allow a fraud incident to pass undetected. Expectedly, the model's computations converge faster at 200 iterations. This study is generic with similar characteristics with other classification methods but distinct parameters thereby minimizing the time and cost of fraud detection by adopting computationally efficient algorithm.

Keywords
Artificial neural network, data mining, detecting fraud transactions, fraud detection and radial basis function network.

1. INTRODUCTION
Internet-based transactions pervade and the associated security issues necessitated rigorous transactions verification to determine their veracity. Financial scam is adversely affecting economies worldwide of which many people have been deprived of substantial amount of valuables. In recent years, the development of new technologies has also provided further ways in which criminals may commit fraud [4]. Individuals, governments, agencies and other business enterprises are deprived of substantial material values, yet detecting and preventing fraud is not a simple task [9].

Fraud is an adaptive crime, so it needs special methods of intelligence gathering and data analysis in order to detect and prevent frauds [27]. Indeed, fraud detection comes to fore once fraud prevention has failed; hence fraud detection must be routinely active in a financial transactions system.

Fraud detection and prevention are concurrent processes—while fraud detection is the **s**potting of false claim, act or data; fraud prevention is the bursting of a false claim, act or data before it materializes, by raising alerts or sending red flags to the relevant dashboards for proactive actions, thus preventing it from occurring. Indeed, fraud detection is triggered once fraud prevention has failed, as fraud detection must be routinely active in a financial transactions system.

Interestingly data mining methods are being harnessed to build models (Test algorithms) to identify and detect the risk of fraud and new techniques are being designed for preventing fraudulent financial reporting [14]. [40] asserts that data mining has capability to extract knowledge from huge data heaps, and have been assisting auditors and crime investigators to detect fraud. More so, as past cases of fraud can be used to build models to identify and detect the risk of fraud [14].

This research work seeks to formulate a model to detect abnormal activities in datasets such as money laundering, electronic commerce scam, dubious insurance, mortgage and health claims, etc. using a radial basis function (RBF) network by creating a series of RBF classification models, select an optimum RBF among the candidate models and evaluate the chosen BRF model that best fits the data using a set of metrics.

Researchers have formulated many concepts, methods and models of detecting and/or preventing fraud however, this is an attempt to use radial basis function, a multivariate interpolation approximation method to classify financial transactions, due to RBF's fascinating features such as its convergence power, easy interpolants formulation, resilience on irregular data distributions and adjustable smoothness render it highly efficient [6].

Financial frauds detection (FFD) in a dataset are generic abnormal activities with similar characteristics but distinct parameters. Notwithstanding the fact that a German bank credit dataset (available online) was used for this experiment, it is believed that the model would exhibit similar behaviour with other localized dataset.

This research would assist governments and private corporate businesses to proactively detect fraudulent business practices, especially as the global economy is in recession, thereby minimizing the time and cost of fraud detection by adopting computationally efficient models rather than mere reliance on intuition, manual long investigations and/or windy auditors' reactive findings.

Fraudulent practices should be checkmated with fraud detection processes embedded in financial transaction systems to be updated at regular fixed time intervals to curb dubious financial transactions.

1.1 Fraud Monitors

If one could isolate the factors that indicate a fraud risk or a high probability of fraud and then develop rules (or controls) and use them to flag only those claims or requests susceptible to be fraudulent. This can identify the symptoms of fraud before large losses occur. Continual routines that monitor key symptoms and track fraud risk trends can also be a major deterrent, thereby preventing or identifying fraud almost as soon as it occurs. To this end, financial regulators often prescribe the use of fraud risk indicators or *red flag* concept [29]. These are fraud symptoms that can be monitored. Various researchers have worked on the concept [2], [16], [21], [22], [24], [25], and [33], their results indicated that fraud risk indicators are the most important factor in fraud detection. Further, red flag method raises auditors' sensitivity to the possibility of fraud [20]. Fraud detection is more of monitoring the behaviour of users' domain in order to estimate, detect, or deter undesirable behaviour [30].

1.2 Data Mining and Fraud Detection

Rather than relying merely on fraud examiners' intuition, data mining techniques are used to combine powerful analytical techniques with known business processes to turn warehoused data into the insight to mitigate the risk of fraud and abuse. Typically, organizations with local or branch offices allow the central office to store and mine data for the entire organization and grant access to local offices. Data mining techniques are replete with various means of developing rules to isolate fraud risks, whereby fraud investigators can identify the symptoms of fraud and raise a red flag. Continual routines that monitor key symptoms and track fraud risk trends can also be a major deterrent, thereby preventing or identifying fraud almost as soon as it occurs.

Hitherto, the traditional data analysis techniques such as regression analysis, cluster analysis, numerical taxonomy, multidimensional analysis, time series analysis, nonlinear estimation techniques, and other multivariate stochastic models have been applied to solving many practical problems, yet they are oriented toward the extraction of quantitative and statistical data elements [36]. Researchers have recently turned to ideas and methods developed in statistics, database technology, artificial intelligence, pattern recognition, machine learning, information theory, knowledge acquisition, information retrieval, high-performance computing, and data visualization.

Due to the dramatic increase in fraud cases which results in loss of substantial worth of valuables globally annually, several modern techniques in detecting fraud are continually developed by researchers and deployed in many business enterprises, where many service agencies have incorporated data mining into their investigating and auditing processes [34].

2. RELATED WORKS

According to [32], fraud detection is important in today's computing environment. The increase in internet usage, electronic commerce and security vulnerability inherent in most systems make fraud detection a topic of interest in modern societies.

The concept of logistic regression for analyzing a dataset in which there are one or more independent variables that determine a binary outcome (a dichotomous variable) is correlated with the study being investigated.

Notable studies in fraud detection include Peer Group Analysis and Break Point Analysis applied to spending behaviour in credit card accounts [4]. Peer Group Analysis detects individual objects that begin to behave in a way different from objects to which they had previously been similar. [19] presented a case study on anti-money laundering (AML) detection, where data mining and natural computing techniques were combined, specifically, *k*-means clustering was used and the value of *k* was determined by AML experts in creating suspicious/unsuspicious groups.

[11] used Fisher's discriminant analysis, fraud detection in credit card operations, the linear nature of Fisher's pattern transformation makes it unsuitable for a problem such as fraud detection, which one hardly expects to be linear rather a nonlinear discriminant analysis (NLDA) neural models was favoured because of its complexity and behaviour with respect to local minima and model training.

[26] presented three-level-profiling for fraud detection. The three-level-profiling method operates at the account level and points to any significant deviation from an account's normal behaviour as a potential fraud. In order to do this, 'normal' profiles were created based on data without fraudulent records (semi supervised). In the same field, [7] also used behaviour profiling for fraud detection.

[10] combined human pattern recognition skills with automated data algorithms for fraud detection. [23], and [39] used the logistic regression method as a tool to discriminate fraudulent actions from legitimate actions for insurance companies and e-commerce. [12] presented a neural network based fraud management technique based on profiling techniques. [13] presented a rule-based tool for fraud detection using a series of machine learning methods.

Some researchers have performed comparisons among the mentioned techniques, for instance, [31] compared logistic regression, neural networks and regression trees. Their results show that the proposed model of neural networks and logistic regression approaches outperform decision tree in solving the problem under investigation. [37] applied neural networks techniques, Gaussian mixture and Bayesian networks, reaching similar results and suggests a combination of multiple methods. Though Bayesian networks are more accurate and require a short training time, they are slower in the application to new instances.

[8] used decision trees (C4.5) and the instance-based learning algorithm to construct early fraud detection methods for classifying fraudsters and legitimate users. This is unsuitable where multiple attributes are being considered.

[38] stated that Machine learning and artificial intelligence solutions are increasingly explored for fraud prediction and diagnosis, especially in insurance domain. Neural network has been widely used due to its ability to model complex and non-linear models, as it does not have any strict limitations and rigorous assumption for the type of input data ([35]; [1]). Its downsides include long learning time, over-fitting error, and black box characteristics ([5]; [17]).

3. METHODOLOGY

The global description of a dataset is the expressive power of the resulting model, which increases as it represents more complex functions, i.e. as we increase the expressive power of a model, we continue to obtain a better fit to the available

dataset. An over simplified model might not be expressive of the data [15]. Machine learning constructs a model with a set of attributes that are trained to produce a model that generalizes well on new data objects, then the model evaluation is carried out to determine how well a model is expected to perform on data objects beyond those in the training set, usually to assess its performance on a separate test set of unseen data.

Succinctly, model training entails a numerical optimization process, and for better understanding of the underlying dataset. Therefore, for a feed-forward model as radial basis function with adjustable centres and non-linear with respect to its parameters, the optimization problem is multivariable nonlinear, and ordinary least squares method is not applicable to estimate its parameters, hence, a set of Gaussian functions with adjustable centres were chosen, such that the complexity of the family of models is increased step-wise, by increasing the number of Gaussians, the number of hidden neurons, etc. Ultimately, one obtains the parameter vector $\boldsymbol{w}$, for which $Y(\boldsymbol{w})$ is the minimum by computing the values of the parameters $\boldsymbol{w}$ in order to minimizing a cost function that mirrors the "distance" between the predictions of the model and the measured values.

A simple linear model with an output $g(x,w)$ with say, N sample training set is:

$$J(w) = \frac{1}{2}\sum_{k=1}^{N}[(y_p(x^k) - g(x^k, w)]^2 \qquad (1)$$

Where x^k is the input variables vector for instance k,
$y_p(x^k)$ is the corresponding measured (target or actual) value of the model,

$\boldsymbol{w}$ is the weight vector, and

$g(x,w)$ is the model output.

The procedure adopted in this work to construct an RBF network for financial fraud classification models is a supervised learning scheme, whereby the network is trained with feature inputs $x_i = (x_{i1},..,x_{ip})$ and the corresponding outputs $y_i \in \{0,1\}$. The sole objective of the training algorithm was to ensure that a set of input features would yield the anticipated set of outputs using the RBF network algorithm; such that the developed final model could subsequently classify previously unseen data features into their respective true classes.

The modeling technique presented in this study was illustrated with *R* codes, a vectorized software, ideal for handling large dataset. A number of *R* family of packages embedded in *R* Using the Stuttgart Neural Network Simulator (RSNNS) library was employed such as *NeuralNetTools*, *Neuralnet*, *Utils*, *ggplot2*, *devtools* etc., which are freely available online at CRAN (*Comprehensive R Archive Network)* repository: https://github.com/cbergmeir/RSNNS.

To realize the RBF process flow, the following procedure was adopted:

1. Train an RBF network with training examples, which consist of a pattern of activities for the input units together with the desired pattern of activities for the output units.
2. In determining how closely the actual output of the network matches the desired output, the weight of each connection was varied so that the network produces a better approximation of the desired output.

Heuristically, attributes selection is performed based on the area of interest, prior to creating a fraud detection model thereby making good models easier to find. We then merge the segregated subordinate attributes to the main attributes. These combined data were fed into the RBF-ANN network for training and the output is stored in the fraud knowledge repository or Detector (see figure 1).

Figure 1 shows data sources spread in multiple locations (Location A, B, C, …, N) usually local and remote data sources, which could be operational, internal or external data sources, which are subsequently aggregated in a single location (a Data Warehouse) via transactions Monitors, using a client-server configuration through TCP/IP protocol. The data are then pre-processed. The selected data is, cleaned and transformed as necessary under the guidance and knowledge of a domain expert (Figure 1, steps 1, 2). 85 per-cent of the data for training and 15 per-cent to test the predictive ability and the generalization of the derived model.

Prior to training, the selected input variables are normalized and used as inputs of an RBF network and train as indicated in the next sections. The sample data was split into two: n_T (training) and n_Q (test) in the ratio 85:15 respectively to construct a classification model based on n_T at different hidden layers and the resulting fitted model was used to classify the n_Q test data. Having obtained several trained (or candidate) models.

3.1 Parameters Initialization

For RBF networks, the parameters initialization is crucial, as they are localized functions; if they are initially located far from the domain of interest, or if their extension (standard deviation or dilation) is not appropriate, training will generally fail [28]. The parameters related to the bias inputs must be initialized to zero, in order to ascertain that the function of the hidden neurons are initialized around zero, so that training starts very slowly.

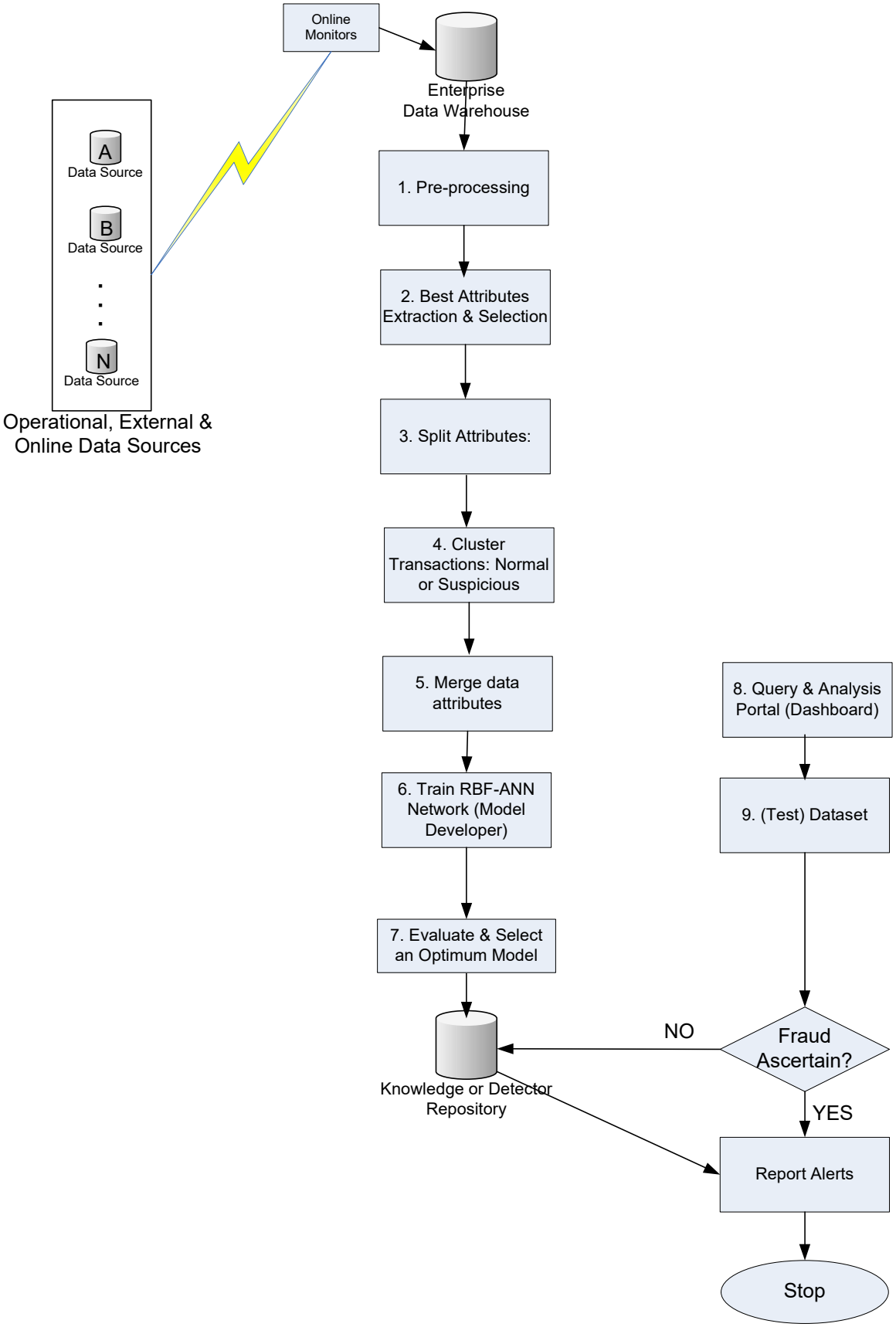


Fig 1: The RBF process flow

As the RBF algorithm was used for base-models creation and classification, with their *R* implementations (Figure 1, step 6), it was simulated with a German Bank credit data. The dataset was randomly partitioned into training and test subsets, to build and evaluate the model's performance. The training and test data ratio split was 85:15 (Figure 1, step 3).

The performance evaluation of the model was derived from several runs and performance metrics generated (Figure 1, step 7). Expectedly, the *German* Bank datasets preliminary results indicated that most of the transactions are normal, while only a few were suspiciously fraudulent.

The RBF network framework also consists of a query portal (a dashboard), where users can verify the status of doubtful transactions, (Figure 1, step 8), the processed (test) dataset, as the case might be is passed on to the trained RBF-ANN fraud-knowledge database to determine its suspicious level (Figure 1, step 9), the level of the transaction suspiciousness would inform the fraud investigator whether it is enough proof of fraud or to probe further.

Finally, the statuses of the transactions are reported and new cases are stored in the Detector repository.

3.2 The Dataset Description

The input variable selection process, i.e., the selection of the components of vector $\boldsymbol{x}$ in $Y(\boldsymbol{x}, \boldsymbol{w})$; can be accomplished in two steps, *viz*:

i) input vector, $\boldsymbol{x}$ dimension reduction, and
ii) the selection of relevant variables whose influence on the response variable is vital rather than the noise variables' influence.

This shows that the input selection process is two prongs, i.e., the model's independent attributes dimension reduction and the rejection of inputs that are irrelevant to the model's prediction.

The study used a dataset of a German Bank Credit data from the UCI Repository of Machine Learning Databases [18]. The dataset is of interest as it consists of a good mix of continuous and nominal attributes. All attributes and values of real identities were removed due to the confidentiality of the data. Records of 1000 bank customers whose 20 features attributes (factors), were found suitable for the model development were used.

The dataset was encoded to include several indicator variables to make it suitable for the RBF algorithm and the categorical response variables anticipated. Several attributes that are ordered categorical have been coded as integer, for instance the predicted response class label $\hat{y}$, was dichotomously defined as follows:

$$\hat{y} = \boldsymbol{W}(\boldsymbol{x}) = \begin{cases} 1, & if\ a\ credit\ trans.\ is\ fraudulent \\ 0, & if\ a\ credit\ trans.\ is\ normal \end{cases} \quad (2)$$

Though, some of the data objects tend to affects the accuracy of the classification model due to outliers and noises that were interpolated by some initial nodes during training, as these could degrade the model's performance. These deficiencies were catered for as described in the subsequent sections.

The data attributes used are:

Table 1: Table of Input Variables Description

S. No	Variables	Description
1.	A1	Status of current Account
2.	A2	Duration
3.	A3	Credit history
4.	A4	Credit purpose
5.	A5	Credit amount
6.	A6	Status of Savings Account
7.	A7	Present Employment Status
8.	A8	Installment rate in percentage of disposable incomes
9.	A9	Gender and Marital
10.	A10	Guarantor(s)
11.	A11	Present Address
12.	A12	Property Ownership
13.	A13	Age (in years)
14.	A14	Other Installment Plans
15.	A15	Housing
16.	A16	Number of existing credits in this bank
17.	A17	Employment Type
18.	A18	Number of Dependents
19.	A19	Telephone
20.	A20	Foreign Worker

3.3 Dataset Normalization

Prior to RBF training, the input variables were normalized and centred: if the inputs have very different orders of magnitude, the smallest ones will not be taken into account during training.

In order to obtain a functional RBF network the following procedure was adhered to:

1. *Find and select the relevant candidate inputs i.e., the input attribute features that are significant predictive factors of the model output,*
2. *Collect the data that is necessary for training and testing the network,*
3. *Determine the appropriate complexity of the model, i.e., the appropriate model parameters,*
4. *Estimate the parameters for which the cost function is minimum, i.e., training the network,*
5. *Evaluate or assess the generalization ability of the neural network after training.*
6. *Iterate procedures (1) – (5) until an optimum model is obtained.*

Note-worthy is the fact that data collection is core in training the desired model as well as the test and validation processes. Hence, the data set must be numerous and sufficient enough to enable the model learn adequately, also, superfluous parameters were avoided to prevent over-fitting of the model; however, too few parameters will hinder the generalization ability of the model, as it might not have learn enough. The selected model net a trade-off between learning and generalization capability by minimizing its cost function.

3.4 Model Creation

To obtain good quality base models, some sufficient and diverse base models could be created either by using different training sets, different deterministic algorithms, different parameter setups, or using randomize or non-deterministic algorithm in conjunction with different training set from the same domain to create each base model; other methods to creating base models by different training sets, viz: *Sampling instances* (drawing multiple training set samples); *Replicating instances* (some instances are selected randomly, to yield different models with unstable models especially); ***Varying instance weights*** (attached to data attributes), *sampling attributes* (using different attribute subsets) and by applying *Attribute transformation* (vary dataset attributes to obtain model diversity).

Since RBF network is considered a stable algorithm as it does not react adversely when its parameters are perturbed, varying the network's weight vector is enough and sufficient to obtain different base models, $m_1, m_2, \ldots, m_h$ using the same training set. Rather than assigning the hidden layer weights randomly, a refined weight adjustment scheme of AdaBoost (i.e. Adaptive boosting) algorithm is apposite, which is particularly suited for a 2-class classification tasks and the error level of the base models does not exceed 5%, thereby enhancing the base models' creation diversity, and prediction quality.

3.5 Model Training Optimization/ Experimental Results and Analysis

The purpose of an acceptable learning process is to produce a model that fits well on a variety of previously unknown data objects and makes the model generalizes better. Machine learning (ML) is the kernel of any data mining technique, due to its capability to gain insight into a problem, data selection and model search criteria. In this study, the optimization techniques were based on the iterative computations, the gradient of the cost function with respect to the parameters of the model. The gradient thus computed is subsequently used to update the values of the parameters found at the previous iteration. (See Table 2).

Table 2: Table of Average Misclassification (Trained) Error Rate (MER) obtained from several RBF neural network runs at different hidden layers and at different RBF iterations. The asterisked (*) indicate the average MER of the best RBF model (i.e., with the least MER) at each RBF iteration run. The best model here is the RBF model with 840 Hidden nodes and 300 iterations and higher.

				Average Misclassification (Train) Error Rates (MERs) in percentage								
	Hidden				Number of rbf Iterations to learn, (r)							Average
S.No.	Nodes (H)	*100*	*200*	*300*	*400*	*500*	*600*	*1000*	*1500*	*2000*	*3000*	MER (%)
1	***10***	68.94	68.94	68.94	69.18	69.18	69.18	69.18	69.18	69.18	69.18	69.11
2	***25***	68.00	68.00	68.12	68.12	68.12	68.12	68.12	68.12	68.47	68.00	68.12
3	***50***	64.83	65.06	65.29	65.53	65.53	65.53	65.65	65.65	65.88	65.29	65.42
4	***80***	62.12	62.82	62.94	63.18	63.18	63.41	63.41	64.00	64.00	63.88	63.29
5	***100***	60.82	61.88	62.35	62.59	62.47	62.59	62.24	62.24	62.24	62.00	62.14
6	***150***	55.77	56.94	57.53	57.53	57.65	57.77	58.24	57.88	57.65	57.41	57.44
7	***250***	47.88	49.53	50.53	50.82	50.94	50.59	50.12	50.12	50.35	50.35	50.12
8	***450***	32.59	38.00	34.47	34.94	34.94	34.71	34.82	36.00	36.00	35.77	35.22
9	***650***	19.29	21.18	21.06	20.94	21.06	20.35	19.53	19.18	19.65	19.88	20.21
10	***750***	12..12	13.88	14.59	14.59	14.59	14.00	12.00	12.24	11.41	11.06	13.15
11	***800***	8.82	10.71	11.41	11.29	11.29	11.06	9.76	9.53	9.53	9.41	10.28
12	***820***	8.00	9.06	9.18	9.77	9.77	9.53	8.94	8.59	8.71	7.88	8.94
13	***830***	7.06	8.24	8.94	9.06	9.41	9.18	9.06	7.65	7.88	7.65	8.41
14	***840***	6.24	***7.29**	***8**	***8.24**	***8.35**	***8**	***7.18**	7.18	7.18	***6.47**	**7.41**
15	***842***	***6.12**	6.82	8.24	8.47	8.94	8.35	8.00	7.65	6.82	6.71	7.78
16	***843***	***6.12**	***7.29**	8.12	8.59	8.47	8.59	8.12	***6.94**	***6.71**	7.53	7.65
17	***845***	6.71	7.65	8.47	9.06	8.82	9.41	8.12	7.76	8.12	8.24	8.24
18	***850***	69.41	69.65	30.24	69.29	30.24	30.24	69.77	51.18	51.18	69.77	54.10
19	***855***	66.82	66.71	65.06	55.41	69.06	30.24	69.77	69.76	63.77	69.77	62.64
20	***860***	67.41	45.06	69.77	69.77	59.65	57.18	69.65	58.35	69.65	69.77	63.63

As stated earlier, the experimentation in this study consists of 1000 records of bank customers on whose 20 features attributes were obtained. The dataset was randomly split as the training set (85%) consists of 850 samples of which 150 samples (15%) were used for the test/validation set using ten (10) different RBF iterations, $r \in \{100, 200, 300, 400, 500, 600, 1000, 1500, 2000, 3000\}$. At each RBF iteration, *r*, twenty (20) RBF models were trained using the training data over a set of twenty (20) different hidden layers, *H* as shown in Table 4.4, i.e., H = {10, 25, 50, .., 860}.

For each model, the class label of the unseen 15% test data was predicted using the fitted RBF network and the performance of the fitted model was assessed using the Average Misclassification Error Rate (MER) stated in equation (3), see Table 3.

Table 3: Table of Average Misclassification (Test) Error Rate (MER) obtained from several RBF neural network runs at different hidden layers and at different RBF iterations. The asterisked (*) indicate the average MER of the best RBF model (i.e., with the least MER) at each RBF iteration run. The best model here is the RBF model with 860 Hidden nodes and 300 iterations and higher.

		Average Misclassification Error (Test) Rates in percentage										
	Hidden	Number of rbf Iterations to learn, (*r*)										Average
S.No.	Nodes (H)	*100*	*200*	*300*	*400*	*500*	*600*	*1000*	*1500*	*2000*	*3000*	MER (%)
1	***10***	71.33	71.33	71.33	71.33	71.33	71.33	71.33	71.33	71.33	71.33	71.33
2	***25***	70.67	71.33	71.33	71.33	71.33	71.33	71.33	71.33	71.33	71.33	71.26
3	***50***	71.33	71.33	71.33	71.33	71.33	71.33	71.33	71.33	71.33	70.67	71.26
4	***80***	66.67	68.67	70.67	71.33	71.33	71.33	71.33	71.33	70.00	69.33	70.20
5	***100***	68.67	69.33	70.67	71.33	71.33	71.33	70.67	70.67	70.67	70.00	70.47
6	***150***	69.33	71.33	71.33	71.33	71.33	71.33	71.33	71.33	71.33	70.67	71.06
7	***250***	68.00	70.67	70.67	70.67	70.67	70.67	70.67	70.67	70.00	70.67	70.34
8	***450***	68.00	70.00	70.00	70.67	70.00	70.67	70.00	70.67	70.00	***68**	69.80
9	***650***	69.33	69.33	68.67	69.33	69.33	69.33	69.33	***69.33**	69.33	68.67	69.20
10	***750***	70.00	68.67	68.67	***68.67**	69.33	68.67	68.67	***69.33**	69.33	69.33	69.08
11	***800***	67.33	68.00	68.67	69.33	***68**	68.67	68.67	68.67	68.67	68.67	68.47
12	***820***	68.37	68.00	68.00	***68.67**	69.33	68.67	69.33	68.67	69.33	69.33	68.78
13	***830***	68.67	69.33	68.67	68.67	68.67	68.67	69.33	68.67	***68.67**	68.67	68.82
14	***840***	70.00	69.33	70.00	70.00	70.00	70.00	70.00	70.00	70.00	69.33	69.87
15	***842***	70.00	69.33	69.33	69.33	69.33	69.33	69.33	69.33	69.33	69.33	69.40
16	***843***	70.00	69.33	69.33	69.33	69.33	70.00	70.00	70.00	70.00	70.00	69.73
17	***845***	***66.67**	68.00	68.67	***68.67**	68.67	69.33	69.33	69.33	69.33	69.33	69.00
18	***850***	71.33	69.33	71.33	70.00	71.33	71.33	***54**	71.33	71.33	71.33	70.96
19	***855***	71.33	**66.00**	**66.67**	70.67	71.33	68.67	68.00	71.33	71.33	71.33	69.67
20	***860***	59.33	72.00	70.67	71.33	71.33	***32.67**	71.33	71.33	71.33	71.33	**66.27**

Consequently, the model with the best average predictive accuracy (i.e., with the least average MER) among the several fitted models are visible at each iteration as indicated by the asterisk (*), such a model becomes the best model at that iteration and the corresponding hidden layer becomes the optimum hidden layer at that run.

4. MODEL SELECTION

The scrutiny of the variation in the prediction error on the validation dataset, n_Q and the iterations stopping criteria, i.e., terminating the RBF training when the prediction error starts diverging. Essentially, the prediction error is estimated as the criterion for model selection, i.e. the estimation of how well the trained model will perform on future unseen datasets, by selecting the model whose estimated prediction error is least, thus yielding the aggregation function:

$$m_*(x) = \frac{1}{h}\sum_{i=1}^{h} m_i(x)\,,$$

where h is the number of hidden nodes. (3)

This is acceptable, more so, as researchers often times tend to aggregate model predictions rather than creating models' representative. The prediction performance of each base model at different hidden layers was evaluated with the average *misclassification error rate* (MER), $\hat{\vartheta}_H$ or the prediction error rate thus:

$$\hat{\vartheta}_H = \frac{1}{S \times n_T}\sum_{r=1}^{s}\sum_{i=1}^{n_T} I(y_T \neq \hat{y}_T) \qquad (4)$$

where, *s* is the cross-validation runs

I(•) ∈ {0,1} is an indicator function

H is the number of hidden nodes

Averaged over the number of cross-validation runs *s*, where *I*(•) I an indicator function whose value is 1 if the predicted class label $\hat{y}_{iT}$ of the ith sample at the *r*th cross-validation run does not equal the true class label y_{iT} of the sample transactions and 0 if otherwise. Hence, $\hat{\vartheta}_H$ is the prediction error rate of the model with *H* number of hidden nodes.

Other model performance evaluation techniques considered were: Confusion matrix, ROC (Receiver Operating Characteristic), Misclassification error, Mean classification error, weighted misclassification error as shown in Table 4, Figure 2 and 3.

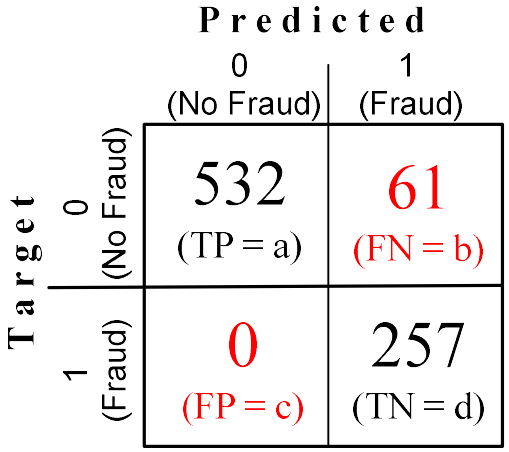
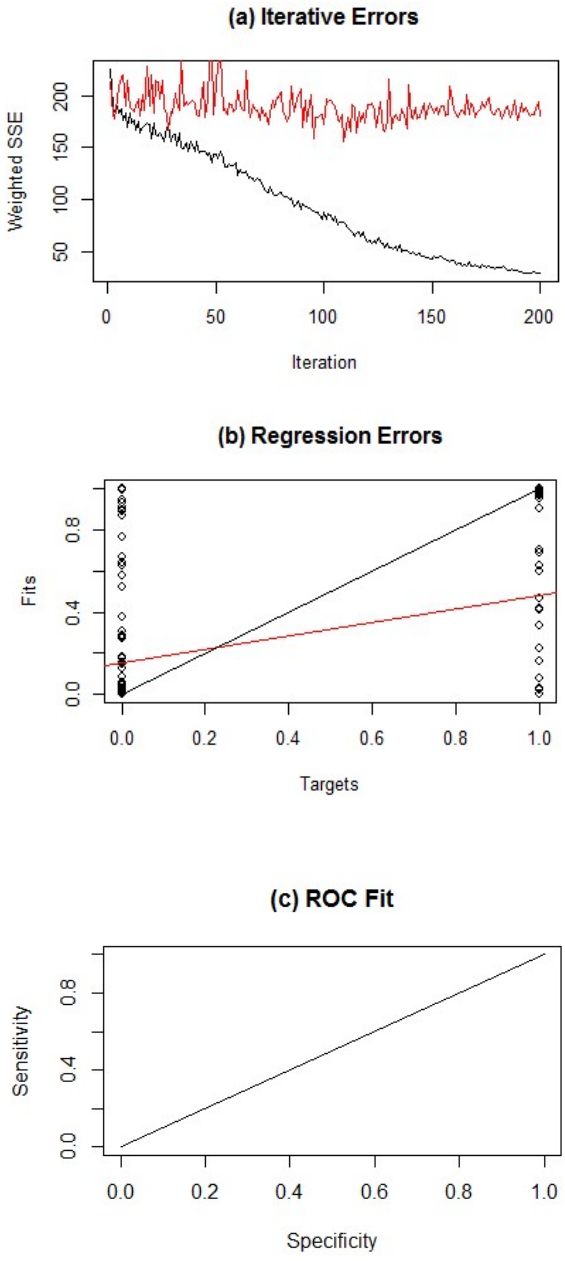


Figure 2: Confusion Matrix

Table 4: Estimates of various performance indices of the optimum RBF model selected for the bank credit transactions

Performance indices of the best trained RBF model	Estimated Values (in %)
Misclassification Error rate	7.18
Prediction Accuracy	92.82
Sensitivity	89.71
Specificity	100
Area under the ROC curve (AUC)	98.6



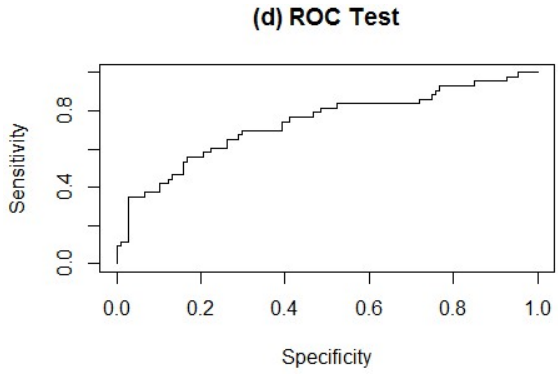


Figure 3: Plots of radial basis function trained with the German credit dataset. (a) The iterative error plot of both training (black) and test (red) error. (b) The regression plot for the test data. As a classification is performed, ideally only the points (0,0) and (1,1) was populated. (c) ROC plot for Specificity against Sensitivity, on the training dataset. (d) Same as (c), but for the test data.

Figure 3a is an Iterative error plot that indicates the summed squared error (SSE), i.e., the sum of the squared errors of all patterns for every epoch (iteration). This is indicative of the quality of the regression. It has target values on the *x*-axis and fitted/predicted values on the *y*-axis. A test set is provided, its SSE is also shown in the plot, normalized by dividing the SSE through the test set ratio (which is the amount of patterns in the test set divided by the amount of patterns in the training set).

The Figure 3b is a regression plot which illustrates the quality of the regression. It has target values on the *x*-axis and fitted/predicted values on the *y*-axis. The optimal fit would yield a line through zero with gradient one. This optimal line is shown, as well as a linear fit to the actual data.

As much as iterative and regression error plots are frequently used for classification problem analysis, nonetheless, the regression error plot is parsimonious with information than for a regression problem therefore, a function for displaying receiver operating characteristics (ROC) is shown (Figure 3c. ROC plots are strikingly appropriate for the analysis of binary classification problems with two classes as the issue at hand (fraud detection). The estimated area under the ROC curve is 0.9860, which is almost 100% coverage of the area under the ROC curve. This is giving more credence to the chosen model as a veritable classification for the dataset under consideration.

Next is the Confusion matrix of the selected model, which shows the number of times the RBF network accurately/erroneously classified a transaction of class 1 (fraud) to be a member of class 0 (no fraud). The outcome of the model as shown in Figure 2, indicates that the model would rather raise alarms where there was none than to allow a fraud incident to pass undetected. This is a very desirable feature for a fraud detection system, whereby the false alarm (of 61 cases) is allowed but having zero-tolerance for fraud incidents is acceptable.

4.1 Comparison with Other Classifiers

Amongst several neural network methods, for brevity only three of these techniques were selected for comparison with RBF. Multi-layer perceptron back-propagation (MLP-BP or MLP for short), Dynamic decay adjustment (DDA) and General nonlinear regression (GNLR) were adopted to benchmark the results of RBF discussed in section 4,

These neural networks techniques are classifiers that are suitable for non-parametric prediction among other common features. The same dataset, *German bank credit data*, was applied to these three algorithms, in order to compare the performance of the neural network techniques, training set 85% and 15% testing.

Table 5: Comparison of classification accuracies (%) (Training) by four neural network techniques. The asterisked (*) indicate the accuracy of the best model for each technique.

	NEURAL NETWORK TECHNIQUES			
No. of Runs	GNLR	MLP	DDA	RBF
1	74.68	77.27	74.19	77.92
2	79.87	76.62	75.22	76.62
3	70.13	70.13	72.03	74.03
4	75.97	70.78	**79.22***	**85.71***
5	**81,82***	75.97	77.27	75.32
6	70.13	70.78	72.08	72.08
7	72.73	**83.97***	76.62	75.97
8	78.57	79.22	77.27	79.22
9	74.68	74.68	76.62	75.32
10	74.03	75.97	74.03	76.62
Average Accuracy	**74.53**	**74.60**	**75.04**	**75.90**

Table 5 shows the performance of the different neural network techniques trained over ten iterations; twenty (20) neural network models were trained using the training data, the average of the ten results of the classification accuracy was used to benchmark the performance of these neural networks. The classification accuracies obtained by these four neural network techniques were compared, with the best accuracy for each technique indicated with an asterisk (*).

The comparative results show that the average accuracies obtained by each neural network technique are somehow similar. The highest average accuracy score obtained was from RBF (75.90%), followed by DDA (75.04%) as expected, while MPL (74.60%) and GNLR (74.53%) were the least. Also, it was noted that RBF and DDA attained the highest accuracy score of (85.71%) and (79.22%) respectively, at iteration 4, while MLP and GNLR attained their highest accuracy at iterations 7 and 5 respectively.

This outcome is expected as the literatures asserted, for instance, RBF has the simplest architecture and thus trained faster, whereas GNLR and MLP require higher attributes, susceptible to noise in order to perform better with a resultant lower computational speed.

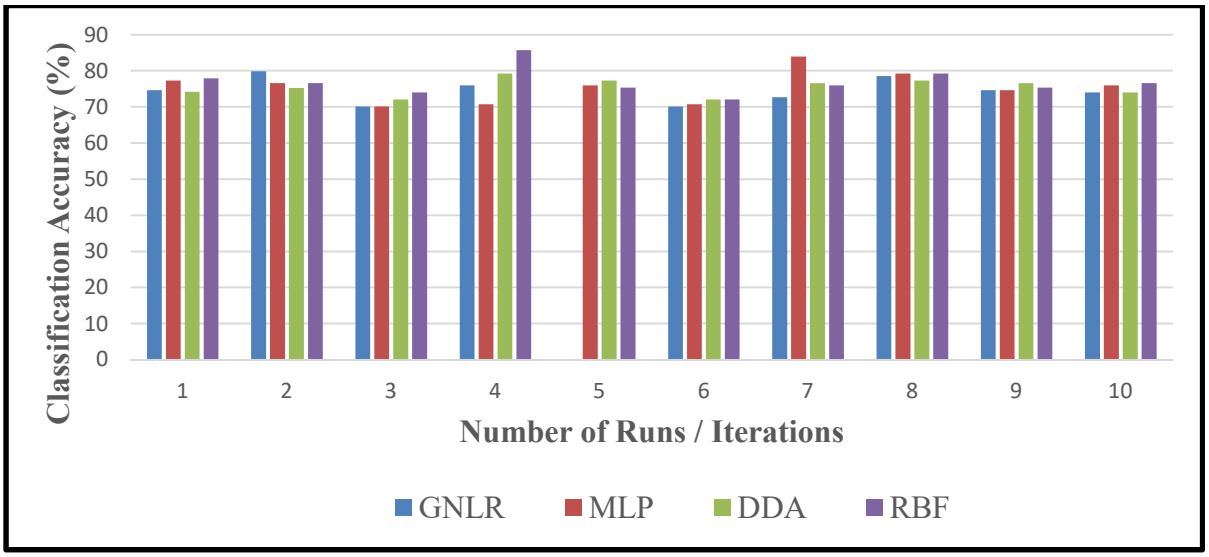


Figure 4: Comparison of neural network classification techniques (training)

The graphical representation of the four neural network techniques comparison is shown Fig. 4. Observe that out of the four classification techniques, RBF has the highest accuracy obtained at a least iteration (4 out of 10) because of its simple structure and fastest computation. Expectedly, next in performance is the DDA technique because of its structural affinity to RBF.

Similarly, the following results were obtained when the test data was applied to the four algorithms:

Table 6: Comparison of classification accuracies (%) (Test) by four neural network techniques. The asterisked (*) indicate the accuracy of the best model for each technique.

	NEURAL NETWORK TECHNIQUES			
No. of Runs	GNLR	MLP	DDA	RBF
1	46.67	47.27	56.69	64.68
2	59.76	46.52	75.22	79.87
3	57.32	59.04	73.32	70.13
4	63.04	67.53	77.22	81.82
5	41,07	55.19	**86.27***	85.97
6	57.36	72.63	72.08	87.13
7	60.13	**79.67***	76.62	**94.73***
8	**71.42***	78.03	72.77	88.57
9	70.16	62.36	78.12	74.68
10	70.45	60.74	44.24	54.03
Average Accuracy	**60.61**	**61.03**	**69.59**	**76.32**

The predictive performance of the different neural network techniques (GNLR, MLP, DDA and RBF) were tested with the 15% dataset on ten iterations; the average of the ten results of the classification accuracy is shown in Table 6. The classification accuracies obtained by these four neural network techniques were compared, with the best accuracy for each technique indicated with an asterisk (*).

The highest average accuracy score obtained was from RBF (94.73%), followed by DDA (86.27%) as expected, while MPL (79.67%) and GNLR (71.42%) were the least. Also, note that RBF and DDA attained their highest accuracy score at iterations 7 and 5 respectively, while MLP and GNLR attained their highest accuracy at iterations 7 and 8 respectively.

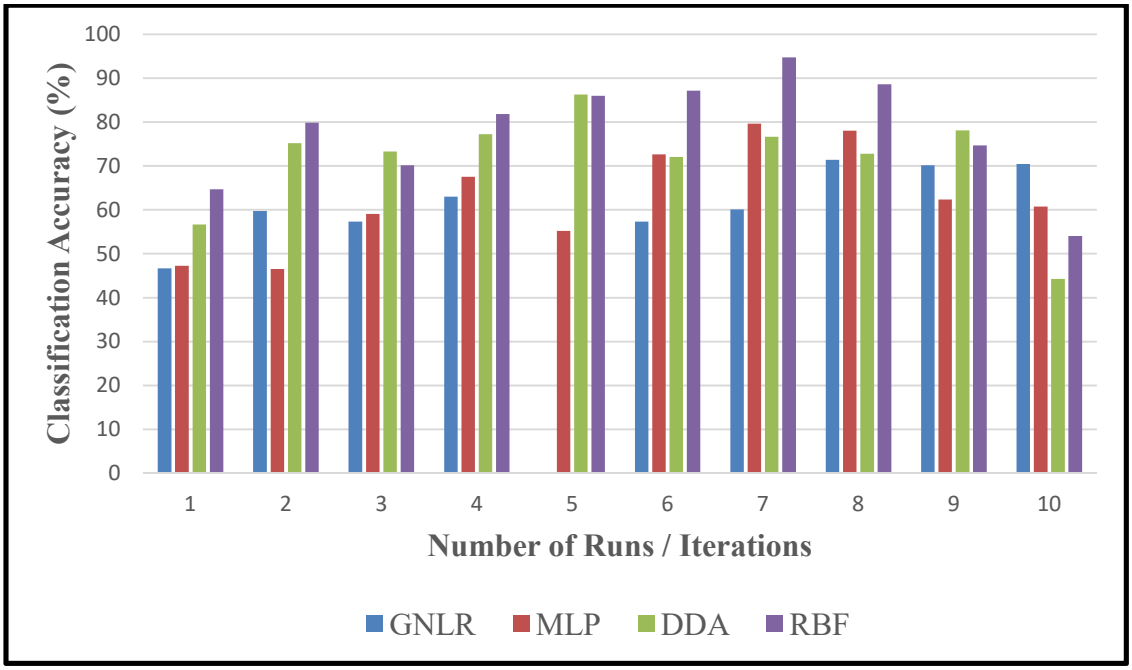


Figure 5: Comparison of neural network classification techniques (test)

The graphical representation of the four neural network techniques comparison is shown Fig. 5. Observe that out of the four classification techniques, RBF has the highest accuracy obtained at iteration 7, which tallies with Table 2, where hidden nodes = 850 at 1000 iterations, equivalent of the seventh iteration. Likewise, the next in performance to RBF technique is the DDA technique for reasons already explained. Hence RBF outperforms the other three techniques, indicating that RBF suggests better prediction in this study.

4.2 Simulation

The execution of these algorithms on an AMD Turion X2 Dual-Core CPU, 2.3GHz, 32-bit on Windows 10 Pro platform took about 180 seconds, this could be mitigated by higher processing power systems.

5. RESULT

Table 2 shows the results of all the fitted RBF networks at twenty different hidden layers, *H* and at different RBF iterations, *r*. The figures in the table indicate the average test sample predictions error rates of a total of 850 bank customers credit transactions constructed on the German bank credit dataset.

At each RBF iteration, *r*, the average MER of the network at different hidden layers H was reported and the average MER of the best model was asterisked. Hence from the results presented in Table 2, the hidden layer that produced the best RBF model can be observed at each iteration. For instance, at 100, 400, 600 and 1500 iterations, the best models were those that have 842, 840, 840 and 843 hidden layers respectively. The corresponding misclassification error rates are: 6.12%, 8.24%, 8% and 6.9% respectively, which translates to an average of about 93% degree of accuracy for the four models.

Worthy of note is the outcome of the Confusion matrix, whereby the False positive (FP) entry returned a value of zero (0), whereas the False negative (FN) entry returned a value of sixty-one (61), see Figure 2, signifying that the model predicted that there was no fraud and actually there was no fraud incident; the model would rather predict there were frauds where there was none than to allow a fraud incident to pass undetected. This is a very salient attraction of a desirable fraud detection system, whereby the false alarm (of 61 cases) is allowed but having zero-tolerance for fraud incidents is desirable. The sixty-one (61) suspected fraud cases in this situation, some of which are misclassified cases, but could be subjected to further scrutiny.

It was also observed in Table 3 that the computations converge easily from 200 iterations and the network become stable, specifically the best hidden layer, 840 yielded about 93% accuracy at 200 RBF iterations. Also, 840 hidden layers yielded the least average MERs of 7.29%, 8%, 8.24%, 8.35%, 8%, 7.18% and 6.47% at 200, 300, 400, 500, 600, 1000 and 3000 RBF iterations respectively. This is affirming the supremacy of the 840 hidden layer at 200 iterations or higher among other hidden layers band it is the appropriate RBF model for the trained data using 200 iterations or higher. However, to conserve computation time, 200 RBF iterations is recommended for the efficient construction of this model since similar prediction outcome would be obtained at higher RBF iterations.

With the Confusion matrix (Figure 2), a number of other performance indices can be computed, but for brevity, the following performance metrics were calculated as follows:

$$\text{Prediction Accuracy} = \frac{a+d}{a+b+c+d}$$
$$\text{Misclassification Error} = \frac{b+c}{a+b+c+d}$$
$$\text{Sensitivity} = \frac{a}{a+b}$$
$$\text{Specificity} = \frac{d}{d+c}$$
$$\text{Negative Predictive Value} = \frac{d}{d+b}$$
$$\text{Precision} = \frac{a}{a+c}$$

In addition to the performance indices given above, many other graphical (visual) and analytic methods are possible it was believed that these few metrics should suffice to justify the formulated classification model to detect fraud practices in financial transactions.

6. CONCLUSION

The most popular model selection strategy (or stopping criterion) in this study was to classify objects and observe the variation of the standard prediction error on a validation set, and in terminating training when the prediction error starts to increase as was the case in Table 3, where the misclassification error begins to increase after 3000 iterations.

Recall that the Training error obtained by applying the model to the dataset by which it was trained, and the Test error (the error obtained from new unseen transaction to the model) test data. The test error is actually how well the model would behave in practice (production) on future data the model had not known. More often than not, Training error always under-estimates Test error drastically, especially in a complex model as this. This explain the divergence in the results of the Training misclassification error rate and Test misclassification error rate as shown in Tables 2 and 3 respectively. However, this disparity can be mitigated (minimized) using a Surrogate convex loss algorithm [3], not discussed here.

7. RECOMMENDATION

RBF networks are feed-forward networks and their activation is not sigmoid as in Multi-layer perceptron (MLP), but radially symmetric (often Gaussian). Thereby, information is represented locally in the network (in contrast to MLP, where it is globally represented). Nonetheless, advantages of RBF networks in comparison to MLPs are mainly, that the networks are more interpretable, training is easier and faster, as better prediction becomes apparent from 200 epoch (see Figure 2). The initialization is performed in the current implementation heuristically vary the RBF weights by a call to successive call to InitFunc (RBF Weights).

As stated earlier, financial frauds are abnormal activities, hence are generic with similar characteristics but distinct parameters. Notwithstanding the fact that a German bank credit dataset was used for this experiment, it is believed that the model would exhibit similar behaviour with other localized dataset. The future work of this study should be able to make some comparisons of the level of frauds in other sub-sectors of the financial industry, such as Export/Import banks, Industrial and Agro-allied development banks where fraud propensity is assumed low.

It is recommended that any fraud detection model must be proactive rather than reactive with zero tolerance for fraudulent practices. A little threshold could imply a significant cost, i.e. if a fraud detection system permits, say 2-in-10 fraud chances, the permissive 2 chances could cost a fortune when a bad 'guy' aimed and strikes appropriately.

Finally, since fraudulent practices are growing in sophistication due to economic contingencies, the researcher recommends that fraud detection processes should be regularly updated at fixed time intervals and fraud detection must be routinely active in a financial transactions system to checkmate dubious criminal tendencies.

7. REFERENCES

[1] Anderson, J.A. & Rosenfeld, E., 1998. Neurocomputing: Foundations of Research. MIT Press, Cambridge.

[2] Apostolou, B., Hassell, J., Webber, S., & Sumners, G., (2001). The relative importance of management fraud risk factors. Behavioral Research in Accounting 13, 1–24.

[3] Ben-David, S., Loker D., Srebro, N., & Sridharam, K., (2012). Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, U.K.

[4] Bolton, R., & Hand, D., (2002). Statistical Fraud Detection: A Review (With Discussion). Statistical Science 17(3): 235–255.

[5] Bishop, C.M., (1995). Neural Networks for Pattern Recognition. Oxford University Press, Oxford, UK.

[6] Buhmann, M. D., (2004). Radial basis functions: Theory and Implementations. Cambridge university press. ISBN 0-521-63338-9.

[7] Burge, P. & Shawe-Taylor, J., (2001). An Unsupervised Neural, Network Approach to Profiling the Behaviour of Mobile Phone, Users for Use in Fraud Detection. Journal of Parallel and Distributed Computing 61: 915–925.

[8] Chang, W. & Chang, J., (2012), An effective early fraud detection method for online auctions. Electronic Commerce Research and Applications 11 (2012) 346–360.

[9] Coderre, D., (2009), Computer-Aided fraud Prevention and Detection, John Wiley and Sons, Inc. Hoboken, New Jersey.

[10] Cox, K., Eick, S. & Wills, G., (1997). Visual Data Mining: Recognising Telephone Calling Fraud. Data Mining and Knowledge Discovery, 1: 225–231.

[11] Dorronsoro, J. R., Ginel, F., Sánchez C., & Cruz C. S., (1997). IEEE transactions on neural networks, 4(8) 827-834.

[12] Field, S., & Hobson, P., 1997. Techniques for telecommunications fraud management. In Proceedings of International Workshop on Applications of Neural Networks to Telecommunications 3, 107–115.

[13] Fawcett, T., & Provost, F., (1997). Combining data mining and machine learning for effective fraud detection. In AAAI Workshop on AI Approaches to Fraud Detection and Risk Management, 14–19).

[14] Gupta, R., & Gill, N. S., (2013), *Prevention and Detention of Financial Statement Fraud – An implementation of Data Mining Framework*. International Journal of Advanced Computer Science and Applications, 3(8), 65-76.

[15] Hand D., Mannila H. & Smyth P. (2001). Principles of Data Mining, A Bradford Book, Massachusetts Institute of Technology Press, London, England.

[16] Hackenbrack K.,(1993). The effect of experience with different sized clients on auditor evaluations of fraudulent financial reporting indicators. Auditing: A Journal of Practice and Theory;1:99–110.

[17] Hippert, H.S., Bunn, D.W., & Souza, R.C., (2005). Large neural networks for electricity load forecasting: Are they overfitted. International Journal of Forecasting, 21, 425–434.

[18] Hoffman, H (2000). UCI Machine Learning Repository [http://archive. .ics.uci.edu/ml]. Institut f"ur Statistik und "Okonometrie Universit"at Hamburg.

[19] Khac, N. L. & Kechadi, M., (2010). Application of data mining for anti-money laundering detection: A case study. 2010 IEEE international conference on data mining workshops. 577-584.

[20] Krambia-Kapardis, M., Christodoulou, C., & Agathocleous, M. (2010). *Neural networks: the panacea in fraud detection.* Managerial Auditing Journal 2010;7: 659–78.

[21] Loebbecke, J.K., Einning, M.M.& Willingham, J. J., (1989). Auditors' experience with material irregularities: frequency, nature and detectability. Auditing: A Journal of Practice & Theory;1:1–28.

[22] Majid, A, Gul, F.A. & Tsui, J., (2001). An analysis of Hong Kong auditors' perceptions of the importance of selected red flag factors in risk assessment. Journal of Business Ethics;3:263–74.

[23] Maranzato, R., Pereira, A., Naubert, M., & Lago, A. P., (2010). Fraud detection in reputation systems in e-markets using logistic regression and stepwise optimization. In ACM SIGAPP Applied Computing Review.

[24] Mock, T.J. & Turner J.L. (2005). Auditor identification of fraud risk factors and their impact on audit programs. International Journal of Auditing 2005;9:59–77.

[25] Moyes, G.L. (2007). The differences in perceived level of fraud-detecting effectiveness of SAS No. 99 red flags between external and internal auditors. Journal of Business & Economics Research; 6:9–25.

[26] Murad, U. & Pinkas, G. (1999). Unsupervised Profiling for Identifying Superimposed Fraud. Proceedings of PKDD'99.

[27] Nigrini, M., (2011). Forensic Analytics: Methods and Techniques for Forensic Accounting Investigations. Hoboken, NJ: John Wiley & Sons Inc. ISBN 978-0-470-89046-2.

[28] Oussar y., & Dreyfus G. (2002). *Initialization by selection for wavelet network training, Neurocomputing*, 34, 131–143.

[29] Pincus KV., (1989). *The efficacy of a red flags questionnaire for assessing the possibility of fraud.* Accounting, Organizations and Society 1989; 14:153–64.

[30] Phua, C., Lee V., Smith, K., & Gayler, R., (2005). A comprehensive survey of data mining-based fraud detection research. Artificial Intelligence Review (2005) 1–14.

[31] Shen, A., Tong, R., & Deng, Y., (2007). Application of classification models on credit card fraud detection. In Proceedings of the 10th International Conference on Service Systems and Service Management,1–4.

[32] Sherly, K.K., Nedunchezhian, R., (2010). Boat adaptive credit card fraud detection system. IEEE (2010).

[33] Smith, M., Omar, N.H., Idris, S., & Baharuddin, I., (2005). Auditors' perception of fraud risk indicators, Malaysian evidence. Managerial Auditing Journal 2005;1:73–85.

[34] SPSS, Statistical Package for the Social Sciences (2000). *Using data mining to detect fraud*. White paper - technical report. SPSS Inc., USA.

[35] Stern, H.S., (1996). Neural networks in applied statistics. *Technometrics 38* (3), 205–216.

[36] Sumathi S. & Sivanandam S. N., (2006). Introduction to data mining and its applications, Studies in Computational Intelligence, Volume 29, ISBN 3-540-34350-4.

[37] Taniguchi, M., Haft, M., HollmTn, J., & Tresp, V., (1998). Fraud detection in communications networks using neural and probabilistic methods. In Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing 2, (pp. 1241–1244).

[38] Viaene, S., Dedene, G. Dedene, & Derring R. A., (2005). Auto claim Fraud detection using Bayesian learning neural networks. Expert systems with applications 29 (2005) 653-666.

[39] Wilson, J. H., (2009). An analytical approach to detecting insurance fraud using logistic regression. Journal of Finance and Accountancy, 1.

[40] Yue, X. Wu, Y. Wang, Y. Li, & Chu C., (2007). A review of data mining-based financial fraud detection research. International conference on wireless communications Sep, Networking and Mobile Computing. 5519–5522.