Jurnal Teknologi

Full paper

Detecting SIM Box Fraud by Using Support Vector Machine and Artificial Neural Network

Roselina Sallehuddin*, Subariah Ibrahim, Azlan Mohd Zain, Abdikarim Hussein Elmi

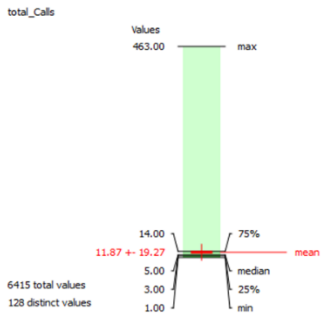
Faculty of Computing, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

*Corresponding author: roselina@utm.my

Article history

Received :12 February 2014
Received in revised form :
15 January 2015
Accepted :15 March 2015

Graphical abstract



Abstract

Fraud in communication has been increasing dramatically due to the new modern technologies and the global superhighways of communication, resulting in loss of revenues and quality of service in telecommunication providers especially in Africa and Asia. One of the dominant types of fraud is SIM box bypass fraud whereby SIM cards are used to channel national and multinational calls away from mobile operators and deliver as local calls. Therefore it is important to find techniques that can detect this type of fraud efficiently. In this paper, two classification techniques, Artificial Neural Network (ANN) and Support Vector Machine (SVM) were developed to detect this type of fraud. The classification uses nine selected features of data extracted from Customer Database Record. The performance of ANN is compared with SVM to find which model gives the best performance. From the experiments, it is found that SVM model gives higher accuracy compared to ANN by giving the classification accuracy of 99.06% compared with ANN model, 98.71% accuracy. Besides, better accuracy performance, SVM also requires less computational time compared to ANN since it takes lesser amount of time in model building and training.

Keywords: SIM box fraud; artificial neural network; support vector machine; classification; accuracy

Abstrak

Penipuan dalam komunikasi meningkat dengan mendadak kerana teknologi moden baru dan lebuh raya komunikasi global, dan dengan itu telah mengurangkan pendapatan dan kualiti perkhidmantan penyedia telekomunikasi terutama di Afrika dan Asia. Jenis penipuan yang paling tinggi ialah penipuan pintasan kotak SIM di mana kad SIM digunakan untuk menyalur panggilan kebangsaan dan antarabangsa daripada operator mobil dan disalurkan sebagai panggilan setempat. Dengan itu amatlah penting untuk mencari teknik yang mampu mengesan jenis penipuan ini dengan berkesan. Dalam artikel ini, dua teknik klasifikasi, Rangkaian Neural Buatan (ANN) dan Sokongan Mesin Vektor (SVM) bagi mengesan penipuan jenis ini. Klasifikasi ini menggunakan sembilan ciri-ciri terpilih daripada data yang diperolehi daripada Pangkalan Data Pelanggan. Perbandingan prestasi di antara ANN dan SVM dilakukan bagi mendapatkan model yang memberikan prestasi yang terbaik. Dari eksperimen yang telah dilakukan, didapati bahawa model SVM memberikan ketepatan yang lebih tinggi berbanding dengan ANN dengan ketepatan klasifikasi 99.06% berbanding dengan ketepatan model ANN, 98.71%. Di samping prestasi ketepatan yang lebih baik, SVM juga memerlukan kurang masa perhitungan berbanding dengan ANN kerana masa pembangunan dan latihan model yang diperlukan adalah lebih rendah.

Kata kunci: Penipuan kotak SIM; rangkaian neural buatan, sokongan mesin vektor; pengelasan; ketepatan

1.0 INTRODUCTION

The theft of service and misuse of voice as well as data networks of telecom providers is considered as fraud. Fraud detection methods continuously evolve from time to time [1]. There is no comprehensive published research in this area mainly due to the lack of publicly available data to perform the experiment. Any broad research published publicly about fraud detection methods will be utilized by fraudsters to evade from detection [2]. The data to be used for the experiments contains confidential information of customers and in most cases law and enforcement authorities prohibit exposing the confidential information of customers, making researchers difficult to access [3-4]. Moreover, many fraud detection problems involved huge data sets that are constantly evolving [5]. For example, data sets can be as large as tenth of thousands of calls per weekday for an organization with three or four thousand employees to hundreds of millions of calls for national carriers. Processing these data sets in a search for fraudster's activities or calls requires more than mere novelty of statistical model, and also needs fast and efficient algorithms.

Existing research work mainly focused on subscription and superimposed types of fraud which are the dominant types of fraud in telecom industries worldwide. However, another type of fraud called SIM box bypass fraud has become a challenging threat to telecom companies in some parts of Africa and Asia. The success of this fraud depends on obtaining SIM cards. Therefore the effects of SIM box bypass fraud vary across countries. In countries where unregistered SIM cards are not allowed and the government laws recognize SIM box devices as illegal equipment, the effect is less compared to countries where obtaining SIM cards by customers is very cheap or even free and government laws do not prohibit unregistered subscribers. The fact that this type of fraud is not a problem for all telecom companies worldwide might justify the reason why the publicly available research on this type of fraud is very limited. SIM box fraud takes place when individuals or organizations buy thousands of SIM cards offering free or low cost calls to mobile numbers. The SIM cards are used to channel national or international calls away from mobile network operators and deliver them as local calls, costing operators' millions in revenue loss [6]. A SIM box is VoIP gateway device that maps the call from VoIP to a SIM card (in the SIM box) of the same mobile operator of the destination mobile.

It can be primarily concluded the major losses associated with telecommunication industry fraud in terms of revenue losses and customer inconvenience is the primary motivation of fraud detection. Efficient fraud detection and analysis systems can save telecommunication operators a lot of money and also help restore subscribers' confidence in the security of their transactions [7-8]. There are different methods available from the literature that has been used to approach different types of fraud in general. These approaches are mainly divided into two subcategories: absolute analysis and differential approach. In both cases, analysis is achieved by means of statistical and probabilistic methods or machine learning methods like decision trees, neural networks, support vector machines etc, applied to the customer information databases like call history, demographic information etc. The existing methods proposed by some of the research work in detecting SIM box fraud are based on high usage and constant activity indicators. However, these features might not be true indicator of this type of fraud currently as the fraudsters are trying to maintain regular usage patterns. On the other hand, those who are using the network excessively might be the very best customers of the company.

The objective of this study is to identify the set of suitable descriptors and appropriate classifier that can be used to recognize SIM cards originated from SIM BOX devices. Neural Networks (MLP) are promising solutions to this type of problem as they can learn complex patterns within a noisy data. Support Vector Machines (SVM) has recently found considerable attention in classification problems due to its generalization capabilities and speed of processing. Therefore SVM will also be applied in this problem. The two classifiers will be compared so that classifier that shows better performance in terms of accuracy and speed can be selected.

The remainder of this paper is structured as follows. The previous research works of fraud detection as well as theoretical implementation of ANN and SVM are briefly reviewed in section 2. Section 3 describes the research methodologies and also the model development for SIM box fraud detection. Section 4 reports the findings and the discussion of the results followed by a conclusion of the study in Section 5.

2.0 RELATED STUDIES

This section reviews some prominent work related to fraud detection methodologies in telecommunication industry as well as other related domains like financial institutions such as banks which have similar fraud characteristics. Most of these approaches are focusing on analysis of the customer information by means of statistical and probabilistic methods, or machine learning algorithms and rule based systems.

Barson *et al.* [9] applied supervised feed-forward neural network (NN) to detect the anomalous use of subscribers. The recent and historic activity profile were constructed and it is found that the empirical results of the system show that NN can accurately classify 92.5% of the subscribers. Krenker *et al.* [10] proves that using bi-directional Neural Network (bi-ANN) in predicting generic mobile phone fraud in real time gave high percentage of accuracy. Bi-ANN is used in prediction the time series of call duration attribute of subscribers in order to identify any unusual behavior. The results show that bi-ANN is capable of predicting these time series, resulting 90% success rate in optimal network configuration. However call duration is the only parameter used, therefore other relevant parameters are missing to accurately predict customer behavior. Farvaresh and Seperi [11] applied decision tree (DT), NN and SVM in order to identify customer with residential subscription of wire line telephone service but used it for commercial purposes to get lower tariffs which is classified as subscription fraud. The employed data mining approach consists of preprocessing, clustering and classification phases. Combination of SOM and K-Means were used in the clustering phase and decision tree (C4.5), Neural Network, SVM as single classifiers were examined in the classification phase. The results are presented in terms of confusion matrix. DT, NN and SVM as single classifiers were able to correctly classify 88.1%, 84.9% and 88.2% respectively. Therefore SVM has shown the best performance among all the classifiers. The limitation might be the computational aspects if implement in real applications.

Neural Networks (MLP) and SVM are promising solutions to fraud detection problem as shown in the above literature discussion. The techniques have been used for different types of fraud in telecom industry as well as financial institutions and they show acceptable results. Therefore SVM and Neural Network were applied in this problem to compare the two classifiers so that classifier that shows better performance in terms of accuracy and speed can be known. Other commonly used classifiers are Naïve Bayes and Decision trees but in this domain they are not widely used and in the cases where they have been used they have not shown good performance results compared to ANN and SVM [11]. The data set that will be used for the experiment contains SIM box fraud SIM cards that have been correctly labelled as fraud and normal SIM cards. This means that supervised learning approach will be used in this classification problem.

2.1 Artificial Neural Network

Artificial Neural Networks (ANN) represents a very basic imitation of the non-linear learning mechanisms of biological neural networks. ANNs have the capability to learn from the environment and enhance their performance through learning which is achieved by an iterative process of adjusting the weights and bias level. A neuron has a number of inputs and one output. It combines all the input values, does certain calculations, and then triggers an output value (activation) [12-13]. There are different ways to combine inputs. One of the most popular methods is the weighted sum, meaning that the sum of each input value is multiplied by its associated weight. Therefore, for a given node g we have:

$$Net_g = \sum w_{ij} x_{ij_t} = w_{0j} x_{0j} + w_{1j} x_{1j} + \ldots w_{ij} x_{ij} \quad (1)$$

Where x_{ij} represents the *i*'th input to node *j*, w_{ij} represents the weight associated with the *i*'th input to node *j* and there are $I+1$ inputs to node *j*.

The value obtained from the combination function is passed to non-linear activation function as input. One of the most common activation functions used by Neural Network is the Sigmoid function. This is a nonlinear function and result in a nonlinear behavior. Sigmoid function is used in this study. The definition of Sigmoid function is as depicted by Equation 2 below:

$$Sigmoid = \frac{1}{1+e^{-x}} \quad (2)$$

Where *x* is the input value and *e* is base of natural logarithms, equal to about 2.718281828. The output value from this activation function is then passed along the connection to the connected nodes in the next layer.

Back-propagation algorithm is commonly used supervised algorithm to train feed-forward networks. The training of the neural network is an iterative process. As each observation from the training set is processed through the network, an output value is produced from the output neuron. This output value is then compared to the actual known value of the target variable for this training set observation, and the error which is the difference between the actual value and the output value is calculated. The weights pointing to the output neurons are modified based on the error calculations. These modifications are then propagated from the output layer through the hidden layers down to the input layer. All the weights in the neural network are adjusted accordingly.

The whole purpose of neural network training is to minimize the training errors. Equation 3 gives one of the common methods for calculating the error for neurons at the output layer using the derivative of the logistic function:

$$Err = O_i\left(1-O_i\right)\left(T_i - O_i\right) \quad (3)$$

In this case, O_i is the output of the output neuron unit *i*, and T_i is the actual value for this output neuron based on the training sample. The error calculation of the hidden neurons is based on the errors of the neurons in the subsequent layers and the associated weights are as shown in equation 4.

$$Err_i = O_i\left(1-O_i\right)\sum_i Err_i W_{ij} \quad (4)$$

O_i is the output of the hidden neuron unit *I*, which has *j* outputs to the subsequent layer. Err_j is the error of neuron unit *j*, and W_{ij} is the weight between these two neurons. After the error of each neuron is calculated, the next step is to adjust the weights in the network accordingly using equation 5.

$$W_{ij,new} = W_{ij} + l * Err_j * O_i \quad (5)$$

Here *l*, is value ranging from 0 to 1. The variable *l* is called learning rate. If the value of *l* is smaller, the changes on the weights get smaller after each iteration signifying slower learning rates. Figure 1 shows the flow of ANN implementation.

2.2 Support Vector Machine

SVM which was developed by Vapnik [14] is based on the idea of structural risk management. SVM is a relatively new computational learning method constructed based on the statistical learning theory classifier [15]. SVM is based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. SVM creates a hyper-plane by using a linear model to implement nonlinear class boundaries through some nonlinear mapping input vectors into a high-dimensional feature space.

For a binary classification problem where there are only two classes in the training data $y_i = \{-1,1\}$ a hyper-plane can be defined as:

$$W.x + b = 0 \quad (6)$$

Where *W* is the normal to the hyper-plane as shown in equation 6 and offset parameter *b* allows us to increase the margin. $|b|/|W|$ is the parameter that determines the shortest distance of the plane from the origin.

For a good classification model, the positive and negative examples of the training data should fulfill the following two conditions:

$$W.x_i + b \geq +1 \text{ for } y_i = +1 \quad (7)$$

$$W.x_i + b \leq -1 \text{ for } y_i = -1 \quad (8)$$

These inequalities can be combined into one set of inequalities:

$$y_i\left(W.x_i + b\right) \geq 1 \forall i \quad (9)$$

The SVM finds an optimal hyper-plane responsible for the largest separation of the two classes. In nonlinear SVM, the training samples are mapped to a higher dimensional space with the help of a kernel function $K\left(x_i, x_j\right)$ instead of the inner product $< x_i, x_j >$. Some of the famous kernel functions are the polynomial kernels, radial basis function kernels, and linear kernels [16]. The equations for these kernels are shown in equation 9, 10 and 11.

Linear Kernel Function: $k\left(x_i, x_j\right) = 1 + x_i^t x_j$ (10)

Polynomial Kernel Function: $k\left(x_i, x_j\right) = \left(1 + x_i^t x_j\right)^p$ (11)

Radial Base Function: $k\left(x_i, x_j\right) = \exp\left(-\partial\left\|1 + x_i^t x_j\right\|^p\right)$ (12)

Where *k* is the kernel function and each data from set x_i has an influence on the kernel point of test value x_j. ∂ is a parameter for RBF kernel and *p* is the number of polynomial degrees for polynomial kernel function. This study considered linear kernel,

polynomial and RBF as a kernel function in SVM model implementation. Choice of kernel functions is the main parameter experimented together with *C* penalty parameter. For each kernel function experimented, the parameters associated with the kernel function that can also have impact on the results are considered.

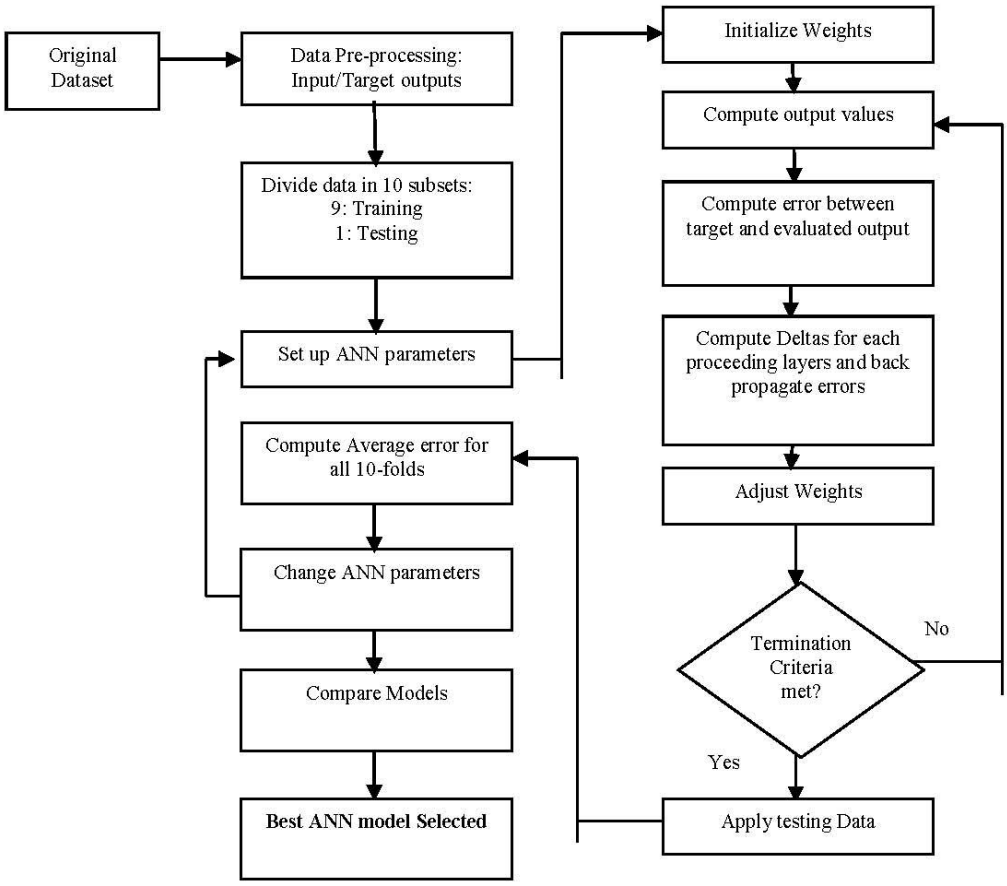


Figure 1 Flow of ANN implementation

■3.0 METHODOLOGIES

This study mainly focused on three different parts, which are the preparation of datasets, development of ANN model and also development of SVM model.

3.1 Preparing Datasets

Here, explanation on preparing the dataset in a format that can be used in the modeling part is given. The first step is to identify the source of data to be used for the models. Then it is followed by data pre-processing that comprises feature extraction, handling missing data, removing outliers and data normalization.

3.1.1 Data Source

The dataset used for the experiments is obtained from the Call Detail Record (CDR) database of a real mobile communication network. Every time a call is placed on a telecommunications network, descriptive information about the call is saved as CDR. CDR contains sufficient information that can be used to analyze the characteristics of each call. The dataset used for the experiments contained 234,324 calls originated by 6415 subscribers from one Cell-ID of the company's network. The dataset consists of 2126 fraud subscribers and 4289 normal subscribers which is equivalent to 66.86% of legitimate subscribers and 33.14% of fraud subscribers.

3.1.2 Feature Extraction and Handling Missing Data

Call Detail Records are not used directly for data mining, since the goal of data mining is to extract knowledge at the customer level, not at the level of individual phone calls. Therefore, the first step in collecting the data is to roll up the different tables and aggregate the data to the required rectangular form in anticipation of using mining algorithms. The data is assembled in the form of columns, with the entity being unique on the row level. Thus, the call detail records associated with customers must be summarized into a single record that describes the customer's calling behavior. All relevant data for each subscriber was assembled in the form of columns and each raw dataset represents a unique subscriber with all the data related to this subscriber included. The choice of features is critical in order to obtain a useful description of the subscriber.

A total of nine features have been identified to be useful in detecting SIM box fraud. Table 1 shows the list of these features and their corresponding data types.

According to Table 1, the features have been selected based on the literature studied on the typical characteristics of SIM Box fraud subscribers as well as contribution from the experience of the staff who work on telecom fraud for the company obtained from the data. In this study, the sample data obtained did not contain any missing values because when sample was taken from the database, filter was applied to only retrieve data that does not contain values which are null.

Table 1 Selected descriptors

	Field Name	Description	Data type
Identification Field	**Call sub**	This is the Subscriber Identity Module (SIM) number which will be used as the identity field	Continuous
Predictor Variables	**Total Calls**	This feature is derived from counting the **Total Calls** made by each subscriber on a single day	Continuous
	Total Numbers Called	This feature is the total different unique subscribers called by the customer (subscriber) on a single day	Continuous
	Total Minutes	Total duration of all calls made by the subscriber in minutes on a single day	Continuous
	Total Night Calls	The total calls made by the subscriber during the midnight (12:00 am to 5: 00 am) on a single day	Continuous
	Total Numbers Called at night	The total different unique subscribers called during the midnight (12:00 am to 5:00 am) on a single day	Continuous
	Total Minutes at night	The total duration of all calls made by the subscriber in minutes at midnight (12:00 am to 5:00 am)	Continuous
	Total Incoming	Total number of calls received by the subscriber on a single day	Continuous
	Called Numbers to Total Calls ratio	This is the ratio of the **Total Numbers Called/Total calls**	Continuous
	Average Minutes	The is the average call duration of each subscriber	Continuous

3.1.3 Identifying and Removing Outliers

Outliers are unusual or abnormal data values and can be errors or real values that are not consistent with most observations. They are usually values that are against the trend of the rest of the data or values that fall near the extreme limits of data ranges. These values were removed from the data to prevent any quality effects it could have on the quality of the models produced. In this study, outlier values were detected using descriptive statistic, graphical method and Z-score standardization.

To identify possible outliers in the data, firstly we need to explore the numerical predictive variables. The descriptive statistics of all the predictive variables are shown in Table 2 and visual representation of the Total Calls variable is shown in Figure 2. The statistics contain the numerical summary measures, including the minimum, maximum and range, measures of mean, median and mode; and measures of variables, such as standard deviation.

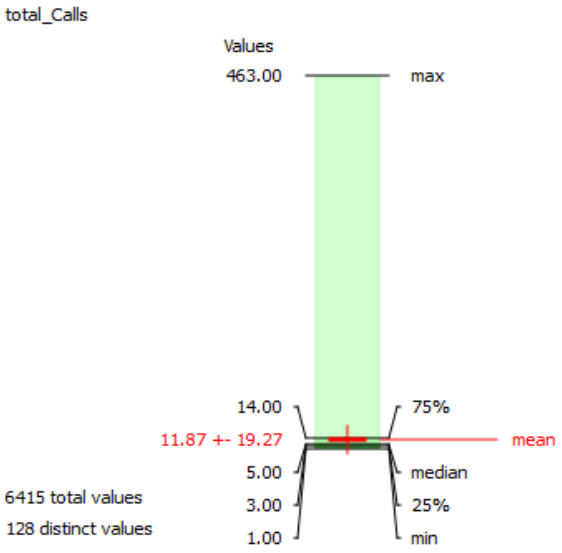


Figure 2 Descriptive statistics of total_call

Table 2 Descriptive statistics of selected features

Descriptive Statistics	TOT_ MIN	TOT_NO CALLED	TOTAL CALLS	TOT_ NIGHT _ CALLS	TOT_ NIGH_ NO_CALL ED	TOT_N IGHT_ MIN	TOT_ INCO M	CALLED /CALLS	AVG_MI N
Mean	23.54	9.83	11.8	2.33	1.46	3.11	1.089	0.73	1.450
Median	4.52	3	5	1	1	0.15	0	0.75	0.88
Mode	21.98	2	3	0	0	0	0	1	0.5
Standard Deviation	45.38	15.97	19.27	3.48	2.35	8.096	2.32	0.27	1.54
Skewness	5.68	6.13	8.57	3.23	3.71	5.33	4.26	-0.49	2.92
Range	875.55	363	462	48	29	128.75	42	0.96	20.39
Minimum	0.0166	1	1	0	0	0	0	0.037	0.0167
Maximum	875.57	364	463	48	29	128.75	42	1	20.41
Count	6415	6415	6415	6415	6415	6415	6415	6415	6415

Most of the variables do not show evidence of symmetry indicating presence of outliers. For example, the Total_Calls variable has a median of 5 and a mean which is double of the median; 11.87. The maximum value is 463 which is an extreme value compared to the average value of subscribers falling in 11 calls. This indicates some right-skewness of the values. Therefore the values in this variable need further analysis of detecting outliers. The rest of the variables are also affected by the values in one of the variable since a subscriber with extreme value in total calls will also have extreme value in duration of the calls or numbers called.

To further validate the existing of outliers in the Total_Call data, the frequency histogram is plot. The histogram shown in Figure 3 is the frequency of the Total Calls variable and the histogram shows right-skewness. There appears to be very small number of values in the extreme right tail of the distributions. The values in these areas are possible outliers. For example, most of the subscribers have calls between 0 and 50 but there are some subscribers having high call numbers more than 400. These subscribers can be considered as outliers that cannot represent the rest of the data.

One of the common methods of using statistics to identify outliers is the Z-Score standardization. Z-score standardization calculate the difference between the field value and the mean of the field and scaling this difference by the standard deviation of the field values as shown in the equation (13) below:

$$Z - Score = \frac{x - Mean(x)}{SD(x)} \tag{13}$$

The data values that lie above the mean will have a positive Z-Score standardization and data value with standard deviation of ± 3 greater or less than the mean is identified as outliers.

The Total Calls variable was applied with the Z-Score Standardization and this method identified 134 subscribers as outliers. The distribution of the value in Total Calls variable now looks better as shown by Figure 4 compared to the original data as shown in Figure 3. All the data was in the left extreme tail and few of data values were on the extreme right tail. Even though the data still shows some right-skewness, this could be explained by the fact the data contained unequal proportion of fraud and normal subscribers. Therefore, since many normal subscribers had small Total Call values, these small values overshadow the fraud subscribers with higher call numbers by skewing the diagram to the right.

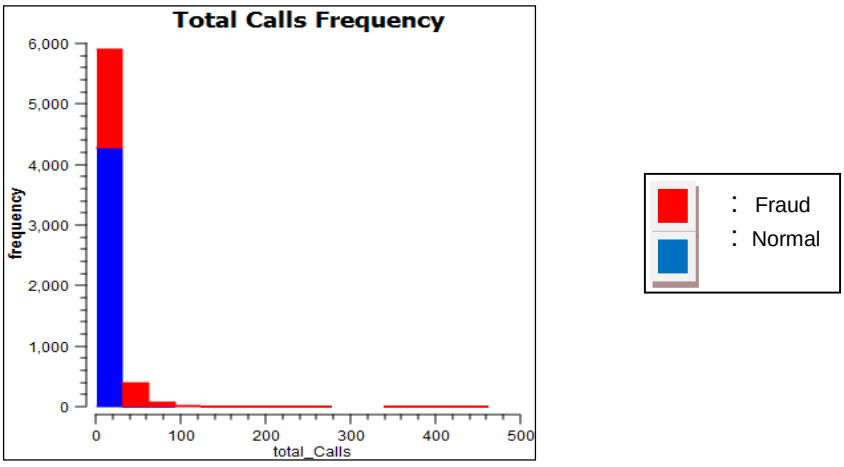


Figure 3 Histogram of total_calls (with outliers)

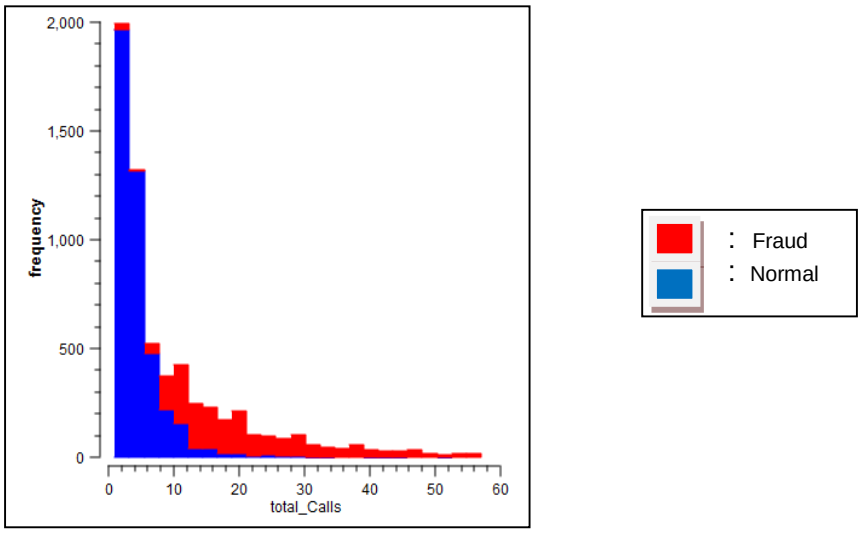


Figure 4 Histogram of total calls attribute with identified outliers removed

After the outliers are removed, data normalization is carried out. All the numerical variables were normalized and compressed to a scale of 0 to 1 to prevent one attribute gives impact to the algorithm's processing power simply because it contains large numbers. Both Neural Network and SVM require attribute to be normalized to this scale. Therefore Min-Max Normalization was applied to the numerical variables. Let *X* refer to the original field value and *N* refer to the normalized field value:

$$N = \frac{X - MIN(X)}{RANGE(X)} = \frac{X - MIN(X)}{MAX(X) - MIN(X)} \quad (14)$$

The data values which represent the minimum for the variable will have a min-max normalization value of 0 and data values representing the field maximum will have a min-max normalization of 1min-max normalization values will range from zero to one.

After the format of the data was prepared according to the requirements of the ANN and SVM modeling techniques, the dataset is partitioned into training dataset which was used to train the models, and testing dataset and validation dataset which was reserved for checking the performance of the models in predicting the target variable.

3.2 Development of ANN Model

To obtain the best Neural Network architecture, four parameter settings were considered. The number of hidden layers in the network architecture as well as the number of neurons in each hidden layer are considered. The learning rate and momentum parameters which have significant effect on the performance of any neural network architecture are also considered.

Different parameter settings were tested until the optimal network architecture is obtained in order to obtain the best ANN model. The number of hidden layers and the number of neurons in each hidden layer have significant influence on the performance of the network [17]. More nodes in the hidden layer increase the power and flexibility of the network for identifying complex patterns. But an overly large hidden layer leads to over-fitting and memorizing the training set. On the other hand, if small number of nodes is involved, it is insufficient to generalize the rules for the training sample. In this research, the determination of number of hidden neurons is referred to Laurene [18] which introduced a rule as stated in equation 15 - 17.

$$h = n \quad (15)$$

$$h = n/2 \quad (16)$$

$$h = 2n \quad (17)$$

Where *n* is the number of neuron in the input layer while *h* is the number of neurons in the hidden layer. In this research, the number of neurons in input layer is 9; therefore 9, 5 and 18 are the number of nodes in the hidden layer.

The back-propagation algorithm is made more powerful through the functionality of a learning rate and momentum term. The learning rate is a constant chosen to help the network weights move toward a global minimum of Sum Square Error (SSE). While momentum helps in the early stages of the back-propagation algorithms, by increasing the rate at which the weights approach the neighborhood of optimality. Experiments with various values of the learning rate and momentum are necessary before the best results are obtained. Therefore, in this research four values of learning rate and momentum which are commonly tried are considered: 0.1, 0.3, 0.6 and 0.9.

To evaluate the models, K-fold Cross-Validation is used to achieve an unbiased estimate of the model performance. In this research, 10-fold cross-validation which is the most common cross-validation technique used for medium sized dataset was applied. Therefore, the dataset was divided into 10 subsets and the model was built 10 times, each time using one out of the subsets for testing and the remaining 9 subsets for training the model. The average error across all the 10 trials is computed. Cross-validation is applied to both SVM and ANN and the models created for each algorithm are compared based on classification accuracy, training duration, precision and recall.

For the ANN development, the nine subsets used for training are combined and network parameters are set. Then random values are assigned to all the weights in the network. The output values are then calculated based on the current weights in the network for each training example. The forward phase finishes with the computation of an error signal. The output errors were

calculated, and the back-propagation process calculates the errors for each output and hidden neuron in the network. The weights in the network are updated. Computing of output values were repeated until the termination criteria is met. The termination criteria used in this research was five hundred iterations of the network. Once, termination criterion is met, the training stops and the network is tested with the set of data points held-out previously. This was repeated 10 times and finally the average errors for all 10-folds were computed. The performance of the model was recorded and different parameters of the network were changed. This process was repeated for all possible combination of parameter settings which was 80 neural network models. The models were evaluated based on their prediction accuracy, generalization error, running time, precision and recall.

From the 80 models that were developed, the prediction accuracy obtained ranges from 56.1% to 98.71%. The details of the best model for each hidden nodes are as stated in Table 3. From Table 3, it can be seen that the model which has two hidden nodes gives the highest accuracy which is 98.71%. The model gives RMSE value of 0.1038. So, the selected neural network model contained four layers. The first layer corresponds to the input values, two hidden layers each having five hidden nodes and output layer with two nodes which represent the two classes: fraud and normal.

Table 3 Details of best model for each hidden nodes

No. of Hidden Layer	Input	Output	RMSE	Accuracy(%)	Time	Precision	recall
1	9	2	0.1055	98.675	13.67	0.987	0.987
2	9	2	0.1038	98.7061	17.17	0.987	0.987
3	9	2	0.1086	98.6906	51.44	0.987	0.987

3.3 Development of SVM Model

Figure 5 illustrates the implementation of SVM model development. There are two basic parameters of SVM model: kernel function and the penalty parameter, *C*. The kernel function in SVM classifier plays an important role of implicitly mapping the input vector into a high dimensional feature space. Common choices of kernel function are the linear kernel, polynomial kernels and radial basis function (RBFs). Linear kernel is a simple kernel function for linearly separable data based on the common inner product plus an optional penalty parameter C. Polynomial kernel is also known as global kernel, it is kernel estimate with two parameters i.e. *C* and polynomial degree *p*. Radial Base Function (RBF), also known as local kernel, is equivalent to transforming the data into an infinite dimensional Hilbert space. This kernel nonlinearly maps samples into a higher dimensional space so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear. *C* and Gamma are two parameters that need to be considered in RBF. To optimize these parameters, 10-fold cross validation was applied to the dataset set and all three kernel functions were experimented. The average error for all ten folds was computed and the performance of each model was recorded.

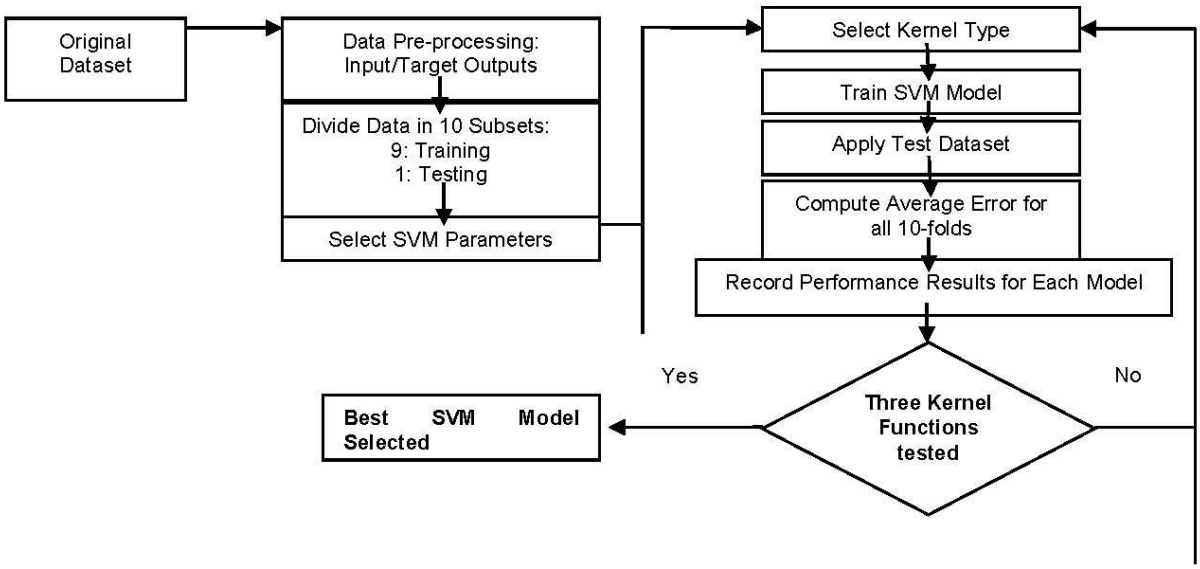


Figure 5 Flow of SVM implementation

Table 4 Details of best model for each kernel functions

Kernel function	Polynomial - Degree	Gamma	C	RMSE	Accuracy (%)	Time	Precision	Recall
Linear/ Polynomial	3		1000	0.1067	98.862	19.71	0.989	0.989
RBF	-	0.125	1000	0.1059	98.8776	5.68	0.989	0.989

The process was repeated for all combinations of three kernel functions and 40 SVM models were developed. The models were evaluated in terms of prediction accuracy, running time, precision and recall in order to find the best SVM model. Table 4 shows the best model for each kernel function. From Table 4, it can be seen that the model that used RBF as a kernel function gives the highest accuracy and lowest RMSE compared to linear and Polynomial, which is 98.8776% and 0.1059. The application of RBF as kernel function increase the capability of SVM convergence time by reducing almost 70% of the time needed in model building. So based on the best performance achieved, SVM model using RBF kernel function with 1000 and 0.125 for gamma and C respectively was selected.

4.0 RESULTS AND DISCUSSION

In this section, a number of evaluations of SVM and ANN models were compared, interpreted and presented. By summarizing the comparison of SVM and ANN models in terms of accuracy, time, and generalization error, it was found that the performance of SVM model for handling the fraud detection is much better than the ANN model. Table 5 illustrates the comparative results obtained from both models.

From Table 5, it can be seen that SVM model gives better performance compared to ANN model. However, cross-validation is the average performance of the models and cannot represent the true performance of the model. To compare the two models, the same parameter settings of the two models were applied with different percentages of training and testing datasets. The dataset have been partitioned into two parts: a training part which is used for training the algorithm and a test part which is used for testing. The percentage of the training and testing was varied in order to study the variations of performance caused by changing the ratio of training to testing partitions of the dataset. For the selection of samples in training and testing portions, the percentage of each class in each portion is preserved. The training and testing portions used contain 10:90, 30:70, 50:50, 70:30 and 90:10.

Then, to compare the classification accuracy of SVM and ANN models, three criteria were chosen which are 1) false negative and false positive rate, 2) classification accuracy evaluation and 3) model building duration evaluation. In the first evaluation, a comparison between false negative rate and false positive rate for both selected SVM and ANN model was conducted.

Table 5 Comparison of selected SVM and ANN model

Features	SVM Best Model	ANN Best Model
RMSE	0.1059	0.10380
Accuracy	98.8776%	98.7061%
Time	5.68	17.17
ROC Area	0.985	0.997
Precision	0.989	0.987
Recall	0.987	0.987

4.1 False Negative Rate and False Positive Rate Evaluation

This section compares the performance of ANN and SVM models in terms of false negative and false positive rate. This comparison was done in order to find which model presents the best performance in terms of identifying the accurate value of normal and fraud subscribers. False positive is the number of normal subscribers that are mistakenly classified as fraud and false negative rate is the number of subscribers that are fraud but mistakenly classified as normal subscriber. Table 6 shows the confusion matrix of SVM and ANN models.

In the confusion matrix shown in Table 6, the columns represent the predicted values and rows represent the actual cases.

In short, the SVM model was able to correctly classify 2074 out of 2126 fraud subscribers and 4269 out of 4289 normal subscribers. Fraud is the negative target value, false negative count is 52 and false positive count is 20. On the contrary, the confusion matrix of ANN model as presented in Table 6, 2063 out of 2126 fraud subscribers was correctly classify. The ANN model also was able to correctly classify 4269 out of 4289 normal subscribers. Since the fraud represents the negative target values, so false negative count for ANN model is 63 and false positive count for ANN model is 20. Hence, it can be concluded that SVM model is better than ANN model. Figure 6 shows the comparison of false negative rate of SVM and ANN.

Table 6 Confusion matrix of SVM and ANN models

	Normal		Fraud	
	SVM	**ANN**	**SVM**	**ANN**
Normal	4269	4269	20	20
Fraud	52	63	2074	2063

Figure 6 compares the false negative rate of the two models. The false negative rate of SVM and ANN were very close when 10 to 50 percent of the dataset was used for training. For both models the false negative rates decline gradually as the training dataset increases. However, the false negative rate suddenly increased dramatically for ANN model when more than 50% of the dataset was used for training and continued to increase as the percentage ratio is increased. On the other hand, the false negative rate continued to decline when SVM model was used and percentage of training dataset was increased to 70%. From this point, false negative rate again started to increase when the training dataset was increased. The minimum false negative rate that could be achieved by ANN was 2.46 when 10% of the dataset was used for training. On the other hand, SVM was able to achieve 1.42 when 70% of the dataset was used for training. After 50% of training dataset, the ANN model performance was degrading significantly to as high as 8.8. Therefore, SVM have shown better performance in correctly classifying fraud subscribers. Figure 7 shows the comparison of false positive rate of SVM and ANN.

The figure shows that the false positive rate decreased steadily to a very low rate when more than 50% of the dataset is used for training. On the other hand, the changes in false positive rate when SVM is used are relatively small as the percentage training data is increased. The figure shows that ANN was able to achieve 0% of false positive rate when 90% of the dataset was used for training the model. On the other hand, the minimum rate achieved by SVM was 0.65 when 10% of the dataset was used for training. This means ANN model has shown better performance in classifying normal subscribers correctly. However, in this research wrong classification of fraud subscribers is more important than wrong classification of normal subscribers because subscribers that are detected as fraud can be further investigated to prove that they are really fraud but fraud subscribers that are classified as normal will remain undetected.

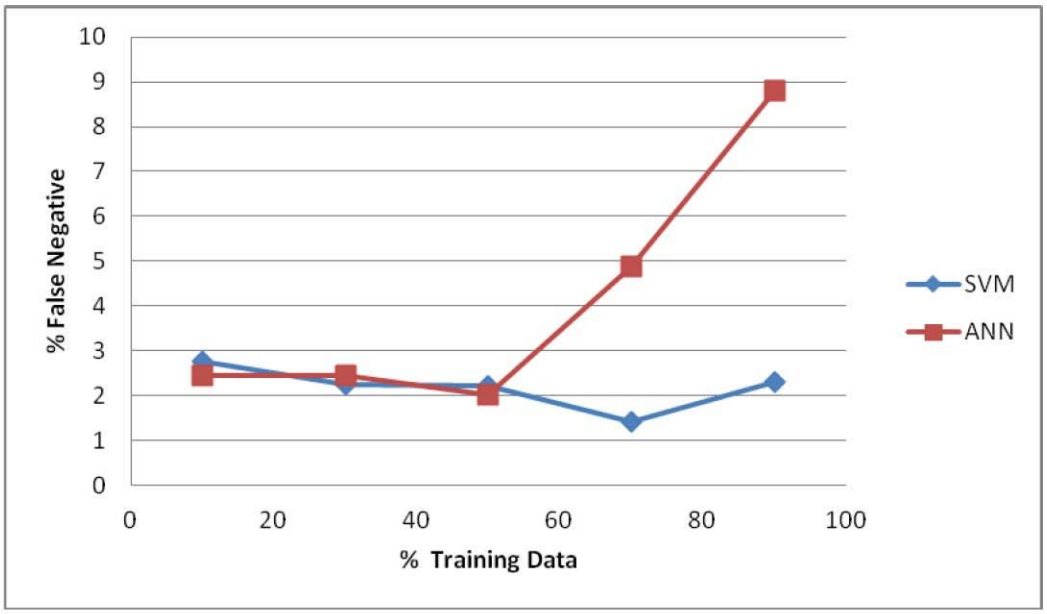


Figure 6 Comparison of false negative rate of the SVM and ANN

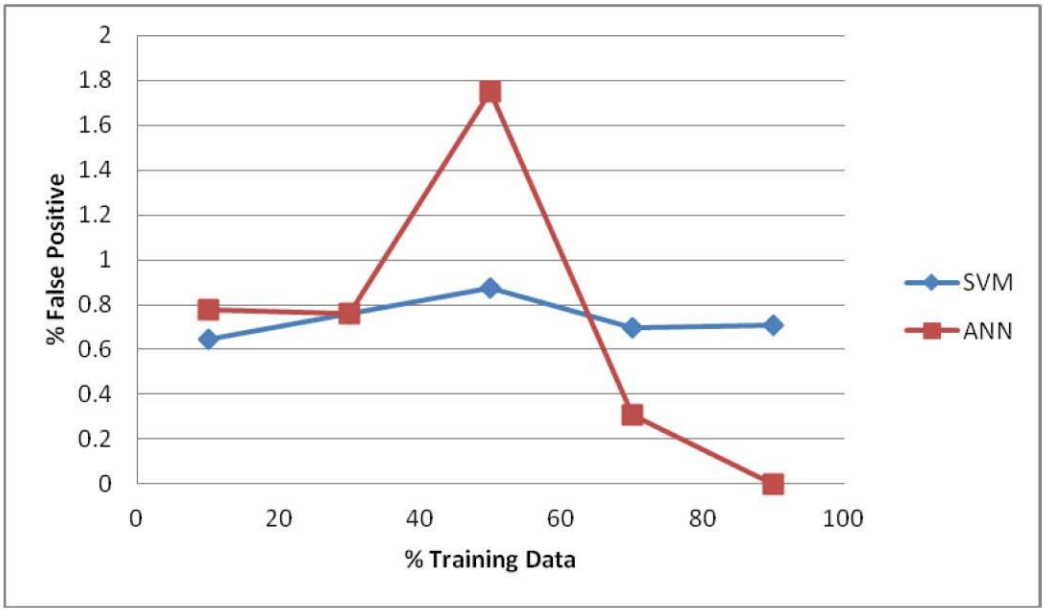


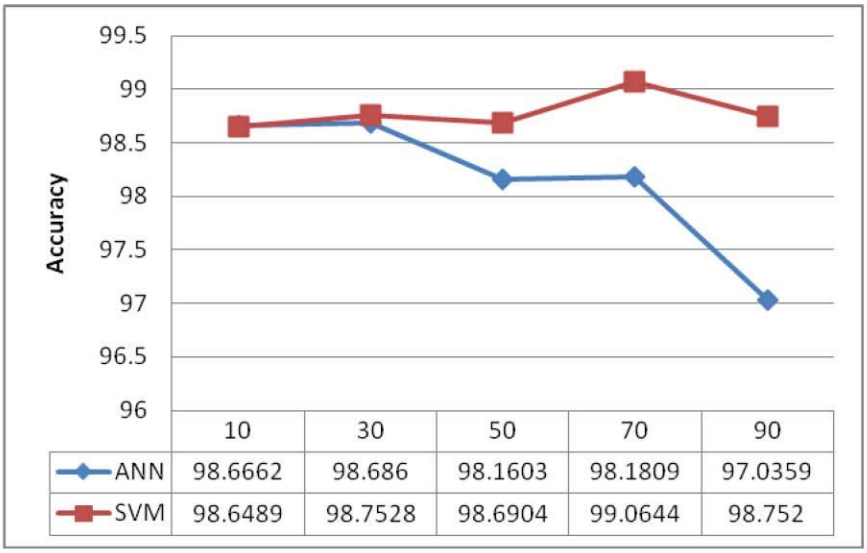
Figure 7 Comparison of false positive rate of SVM and ANN

4.2 Classification Accuracy Evaluation

A comparison of the classification accuracy for SVM and ANN model was also compared. Classification accuracy of SVM is compared with ANN model in order to measure the percentage of subscribers that were correctly classified. 10%, 30%, 50%, 70% and 90% of the dataset were used for training the SVM and ANN. Figure 8 compares the overall classification accuracy of the two models with 10-fold cross validation.

The highest classification accuracy achieved by ANN model was 98.686% when 30% of the dataset was used for training the model. From this point, the performance of the model was degrading and finally when 90% of the dataset was used for training, the model has shown 97.04% classification accuracy. On the other hand, the performance of the SVM model was improving when 10 to 70% of the dataset was used for training the model. ANN requires less training data and more testing data while the opposite is true for SVM.

The highest classification accuracy achieved by SVM was 99.064% when 70% of the dataset was used for training, which compares the ANN about 0.3789%. The results obtained show that SVM have shown better performance than ANN model in terms of classification accuracy. In a word, SVM shows better performance in fraud detection compared to ANN in terms of classification accuracy.



	10	30	50	70	90
ANN	98.6662	98.686	98.1603	98.1809	97.0359
SVM	98.6489	98.7528	98.6904	99.0644	98.752

Figure 8 Comparison of SVM and ANN classification accuracy

4.3 Model Building Duration Evaluation

For extensive comparison, the building and training duration of SVM and ANN mode was compared in Figure 9. This evaluation is conducted in order to show the effect of sample size to the model since model building duration is the running time of training process

From Figure 9 it can be seen that SVM takes less time in model training and development. It can be clearly seen that SVM took more than three times less than the time taken by ANN model in building and training the model. For example, when 70% of the training dataset was used to build the model, it has taken only 5.55 seconds for SVM, while it has taken 16.64 seconds for ANN model. This is more than three times the duration taken by SVM. SVM takes less time in building and training the models also means that SVM requires less computational power compared to ANN.

5.0 CONCLUSION

The focus of this research was to come up with a set of features that can be used to effectively identify SIM cards originating from SIM box devices and an algorithm that can classify subscribers with high accuracy and less computational power. The learning potentials of neural network and support vector machine for the detection of SIM box fraud subscribers were investigated. A total of nine features found to be useful in identifying SIM box fraud subscriber are derived from the attributes of the CDR. The selected features are Total Calls, Total Numbers Called, Total Minutes, Total Night Calls, Total Numbers Called at Night, Total Minutes at Night, Total Incoming Calls, Called Numbers to Total Calls Ratio and Average Minutes. All possible combination of the parameters has been experimented and 10-fold cross-validation has been used to test and evaluate the performance of ANN and SVM models developed. The best model based on classification accuracy, time, generalization error and precision is selected among 80 ANN and 40 SVM models developed. The two best ANN and SVM models are then compared by applying with various partitions of training and testing datasets. The effect of parameters are also analysed in both ANN and SVM models. Experimental results revealed that SVM has better accuracy compared to ANN. SVM gave 99.06% accuracy and ANN gave 98.69% accuracy. The results also show that ANN has higher false negative rates and takes three times more than the time taken by SVM in model building and training. Therefore SVM approach is more appropriate for use in classification model for SIM BOX fraud detection. However, hybridization method between SVM and other computational approach such as particle swarm optimization (PSO) or genetic algorithm are expected to improve the capability of the developed SVM model.

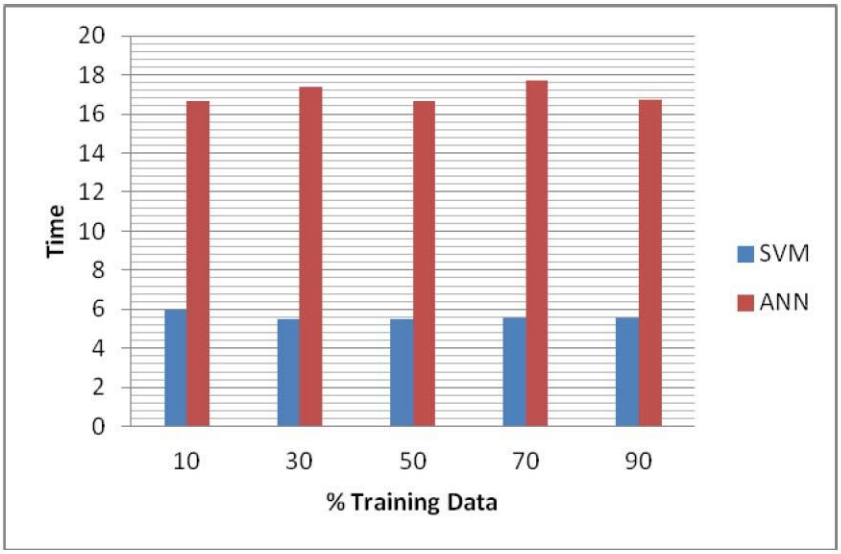


Figure 9 Comparison of SVM and ANN model training duration

Acknowledgement

Special appreciation goes to the reviewer(s) for useful advices and comments. The authors greatly acknowledge the Minister of Higher Education (MOHE) and Research Management Centre, UTM for financial support through the ERGS No. R.J130000.7828.4L107 and GUP No. Q.J130000.2628.08J02. Special thanks also go to Soft Computing Research Group (SCRG) and Information Assurance and Security Research Group (IASRG) for the support and motivation in making this study a success.

References

[1] Zou K., Sum W., Yu H. and Liu F. 2012. ID3 Decision Tree in Fraud Detection Application. *International Conference on Computer Science and Electronics Engineering*. 399–402.
[2] Azgomi N.L. 2009. A Taxonomy of Frauds and Fraud Detection Techniques. *Proceeding of CISTM 2009 Ghaziabad: India*. 256–267.
[3] Hilas, C. and Sahalos, J. 2005. User Profiling for Fraud detection in Telecommunication Networks. *5th International Conference on Technology and Automation*. 382–387.
[4] Suman and Nutan. 2013. Review Paper on Credit Card Fraud Detection. *International Journal of Computer Trends and Technology*. 4(7): 2206–2215.

[5] Bolton, R. J. and Hand, D. T. 2002. Statistical Fraud Detection: A Review. *Statistical Science*. 17: 235–249.
[6] Telenor, G. S. 2010. *Global SIM Box Detection*.
[7] Hynninen J. 2000. Experience in Mobile Phone Fraud.
[8] Phual C., Lee V., Smith K. and Ross G. A. 2007. A Comprehensive Survey of Data Mining-based Fraud Detection Research. *Artificial Intelligent Review*.
[9] Barson P., Field S., Davey N., McAskie G. and Frank R. 1996. The Detection of Fraud in Mobile Phone Networks. *Neural Network World*. 6(4): 477–484.
[10] Krenker A., Volk, M., Sedlar, U., Bester, J. and Kos, A. 2009. Bidirectional Artificial Neural Networks for Mobile-phone Fraud Detection. *Etri Journal.* 31(1): 92–94.
[11] Farvaresh, H. and Sepehri, M. M. 2011. A Data Mining Framework for Detecting Subscription Fraud in Telecommunication. *Engineering Applications of Artificial Intelligence*. 24(1): 182–194.
[12] MacLennan, J. 2009. *Data Mining with Microsoft SQL Server*. 2008. Wiley Publishing Inc: Indianapolis.
[13] Mark E.M. and Venkayala S. 2007. *Java Data Mining Strategy, Standard, and Practice*. San Francisco: Diane D.Cerra.
[14] Paliwal M. and Kumar U.A. 2009. Neural Networks and Statistical Techniques: A Review of Applications. *Expert System with Application*. 36: 2–17.
[15] Vapnik V. 1995. *The Nature of Statistical Learning Theory*. Springer. New York.
[16] Chiu, N. H. and Guao, Y. Y. 2008. State Classification of CBN Grinding with Support Vector Machine. *Journal of Material Processing Technology*. 201: 601–605.
[17] Radhika Y. and Shashi M. 2009. Atmospheric Temperature Prediction using Support Vector Machines. *International Journal of Computing Theory Eng*. 1: 1793–8201.
[18] Notton, G., Paoli, C., Ivanova, L., Vasileva, S. and Nivet, M. L. 2013. Neural Network Approach to Estimate 10-min Solar Global Irradiation Values on Tilted Planes. *Renewable Energy*. 50: 576–584.