



Application of Fuzzy K-Nearest Neighbour on Anomaly Detection System in Customers' calls profiles.

¹Ismaila W. Oladimeji, ¹Adeosun O. Olajide, ²Ismaila Folasade. M., ¹Ogunjinmi T. Omotayo

¹ Department of Computer Science and Engineering

Ladoke Akintola University of Technology, Ogbomoso, NIGERIA,

²Department of Computer Science, Osun State Polytechnic, Iree, NIGERIA

Abstract: Anomaly detection in system calls tends to pose significant solution to abnormal behaviours caused by intruders. System call fraud is the most prominent form of anomaly in mobile device experienced by telecommunication industries. Main problems of IDS are the number of generated alerts, alert failure as well as identifying attack types. This paper aims at implementing Fuzzy K-Nearest Neighbour algorithms on telecommunication customers' calls profile to detect anomaly or intrusion based on an off-line analysis.

Three hundred customers' calls data obtained online were used. The data was normalized and relevant features were extracted using Principal Components Analysis (PCA). Classification tasks were performed by Fuzzy K-Nearest Neighbour (FKNN) algorithm in the MATLAB environment. The system performance was evaluated using Specificity, Sensitivity, and Classification Accuracy. The results showed that the system produced averages sensitivity, specificity, accuracy of 85.6%, 72.0%, 92.3% and 88.2% respectively for FKNN with PCA, while it produced 60.92%, 65.11% and 64.51% for sensitivity, specificity, accuracy respectively for FKNN without PCA. These results are better than that of KNN with PCA of 62.2%, 70.0% and 76.0% for sensitivity, specificity, accuracy respectively and sensitivity of 47.2%, specificity of 50.1%, and accuracy of 55.2% for KNN without PCA.

Keywords: FKNN, Intrusion Detection, Telecommunication, PCA, Cross Validation

I. Introduction

The emergence of Telecommunication has brought a new era in communication industry. The internet, mobile phone and computer have brought about a fundamental shift in pattern of communication and human relationship, amazing social, economic, cultural and psychological transformation. With the growth of network technology and communication, nowadays people are learning various ways to attack through the network resources and carry out extremely destructive attacks. In recent years, the amount of hackers is growing 10 times per year therefore; security as becomes a vital factor of computer technology in telecommunication. [12] In the telecommunication industry, fraud constitutes itself as a major threat to profit margins, not only does it mean less revenue for not paid services but can also increase direct or indirect costs. If not properly addressed, fraud has become a critical issue to all telecom providers. The telecommunications industry generates and stores huge amounts of data regarding calls, SMS (Short Messages Service), MMS (Multimedia Messaging Service) and Internet services of clients. Due to such an amount of transactions, only automated intrusion detection systems (IDS) have enough power to skim over these data and select cases of possible anomalies. As there is no way to know the intention of people behind each of the transactions, algorithms must check for signs of fraud. [1]

The term intrusion is used to explain the process of gaining unauthorized access to a system. The concept of Intrusion detection system (IDS) is useful to detect, identify and track intruders. It is the process of monitoring and analyzing the events occurring in the network in order to detect signs of security problem as very large amount of sensitive information are stored and processed in systems across the globe. Some of the well documented ways of attacking a system [2] are Probes and Scans, Privilege Escalation, Denial of Service, Packet Sniffers, Masquerading, Trojan Code, Viruses, and Worms.

However, many researches have being carried out to curb the menace of attacks in telecommunication which involved the use of algorithms like neural networks, support vector machines, linear discriminant analysis, KNN etc. These algorithms could not function optimum because of their individual deficiencies especially KNN. Thus, this paper employed the hybridized fuzzy and KNN to further strengthen the prevention of telecommunication anomalies.

The FKNN classifier system possesses the capability to adjust itself to the ambiguity in the training and testing data. The classifier ascribes a membership value to the unlabelled object and this provides the system with information which is useful in estimating the accuracy of the decision. This fuzzy membership value identifies what fraction of an unlabelled object resides in each of the defined classes. A benefit of using a fuzzy system for

classification is that an object can be attributed to the category *unknown*, which may provide a more advantageous over crisply assigning the example to the wrong category. Another advantage is that the FKNN classifier training and validation sets can be quickly maintained over time can be modified often and can function with few examples for each class.

II. Related Works

The proliferation of intrusions of fraud in telecommunication have sparked researches by both academics and proprietaries to find solutions to curb the menace. Anderson [16] introduced the concept of intrusion detection in 1980 and a model based on this was also introduced by Denning [17], The researchers in [3] used formal program specification language to create a rule base, the source code of the program was used to identify all its legal actions and used this to create definitions of normal. Clearly a lot of effort has been directed to the use of system call analysis in the area of anomaly detection. It has been established conclusively that system call analysis can reveal the details about program behavior. The authors in [4] implemented an intrusion detection system based on the system call approach but variable length patterns was used in creating profiles unlike Forrest's original work of [5]. In paper [8], the authors prototyped an intrusion detection system that used traces sequences of system calls to identify normal program behavior. In the approach, the intrusion detection system was made to synthetically learn about normal program behavior, during monitoring any different sequence of system calls observed, an anomaly was flagged. [7] took a slightly different approach to using system calls, Instead of keeping track of sequences of calls; the frequency of system calls that are issued by a program was monitored. For more related information see [9, 10]. The writers in [8] introduced a hybrid learning approach of combining k-medoids clustering and naive bayes classification, because of the fact that k-medoids technique represents the real world scenario of data distribution. The algorithms group the whole data into clusters more accurately than K-means such that it results in better classification. The paper [11] introduced the application on intrusion detection based on K means clustering algorithm is used for intrusion detection to detect unknown attack and partition large data space effectively but it has many disadvantages degeneracy cluster dependence. [12] explained the process of intrusion detection which is the major part of network activity and security policies adopted over the network to secure it, four intrusion detection approaches which include Artificial Neural Network, SOM, Fuzzy Logic and SVM were used for the research. [13] developed an intrusion detection based on K-means clustering and OneR classification approach was proposed which combines the techniques of K-means and OneR. In this paper, fuzzy K Nearest Neighbor algorithm will be focused on.

III. The Theory of Algorithms

A. Principal Component Analysis

Principal Component Analysis (PCA) is a dimension-reduction tool that can be used to reduce a large set of variables to a small set that still contain• Principal component analysis (PCA) is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible as most of the information in the large set. Principal component analysis (PCA) has proven to be an exceedingly popular technique for dimensionality reduction and is discussed at length in most texts on multivariate analysis. Its many application areas include data compression, image analysis, visualization, pattern recognition, regression and time series prediction. [18]

In PCA Algorithm, the following are steps involve;

Step 1: Column or row vector of size N_2 represents the set of M images ($B_1, B_2, B_3 \dots B_M$) with size $N \times N$

Step 2: The training set image average (μ) is described as

$$\mu = \frac{1}{M} \sum_{n=1}^M B_n \quad (1)$$

Step 3: the average image by vector (W) is different for each trainee image

$$W_i = B_i - \mu \quad (2)$$

Step 4: Total Scatter Matrix or Covariance Matrix is calculated from Φ as shown below:

$$C = \sum_{n=1}^M W_n W_n^T = A A^T \quad (3)$$

where $A = [W_1 W_2 W_3 \dots W_n]$

Step 5: Measure the eigenvectors U_L and eigenvalues λ_L of the covariance matrix C .

Step6: For image classification, this feature space can be utilized.

Measure the vectors of weights

$$\Omega T = [w_1, w_2, \dots, w_M], \quad (4)$$

whereby,

$$H_k = U_k^T (B - \mu), \quad k = 1, 2, \dots, M' \quad (5)$$

PCA's key advantages are its low noise sensitivity, the decreased requirements for capacity and memory, and increased efficiency given the processes taking place in a smaller dimensions. [20].

B. K-Nearest Neighbor

An instance based learning method called the K-Nearest Neighbor or K-NN algorithm has been used in many applications in areas such as data mining, statistical pattern recognition, image processing. Successful applications include recognition of handwriting and satellite image.

Definition

Suppose each sample in our data set has n attributes which we combine to form an n -dimensional vector:

$$x = (x_1, x_2, \dots, x_n). \quad (6)$$

These n attributes are considered to be the independent variables. Each sample also has another attribute, denoted by y (the dependent variable), whose value depends on the other n attributes x . We assume that y is a categorical variable, and there is a scalar function, f , which assigns a class, $y=f(x)$ to every such vectors. We do not know anything about f (otherwise there is no need for data mining) except that we assume that it is smooth in some sense. We suppose that a set of T such vectors are given together with their corresponding classes: $x^{(i)}, y^{(i)}$ for $i=1, 2, \dots, T$. (7)

This set is referred to as the training set. The problem we want to solve is the following. Supposed we are given a new sample where $x=u$. We want to find the class that this sample belongs. If we knew the function f , we would simply compute $v = f(u)$ to know how to classify this new sample, but of course we do not know anything about f except that it is sufficiently smooth.

The idea in k-Nearest Neighbor methods is to identify k samples in the training set whose independent variables x are similar to u , and to use these k samples to classify this new sample into a class, v . If all we are prepared to assume is that f is a smooth function, a reasonable idea is to look for samples in our training data that are near it (in terms of the independent variables) and then to compute v from the values of y for these samples.

When we talk about neighbors we are implying that there is a distance or dissimilarity measure that we can compute between samples based on the independent variables. For the moment we will concern ourselves to the most popular measure of distance: Euclidean distance. The Euclidean distance between the points $(x_1, x_2, \dots, x_p)$ and $(u_1, u_2, \dots, u_p)$ is $\sqrt{(x_1 - u_1)^2 + (x_2 - u_2)^2 + \dots + (x_p - u_p)^2}$.

The simplest case is $k = 1$ where we find the sample in the training set that is closest (the nearest neighbor) to u and set $v = y$ where y is the class of the nearest neighboring sample. It is a remarkable fact that this simple, intuitive idea of using a single nearest neighbor to classify samples can be very powerful when we have a large number of samples in our training set. It is possible to prove that if we have a large amount of data and used an arbitrarily sophisticated classification rule, we would be able to reduce the misclassification error at best to half that of the simple 1-NN rule.

For k-NN we extend the idea of 1-NN as follows. Find the nearest k -neighbors of u and then use a majority decision rule to classify the new sample. The advantage is that higher values of k provide smoothing that reduces the risk of over-fitting due to noise in the training data. In typical applications k is in units or tens rather than in hundreds or thousands. Notice that if $k=n$, the number of samples in the training data set, we are merely predicting the class that has the majority in the training data for all samples irrespective of u . This is clearly a case of over-smoothing unless there is no information at all in the independent variables about the dependent variable.

There are two difficulties with the practical exploitation of the power of the k-NN approach. First, while there is no time required to estimate parameters from the training data (since the method is not a parametric one) the time to find the nearest neighbors in a large training set can be prohibitive.

C. Fuzzy K-Nearest Neighbour

The Fuzzy K-Nearest Neighbour (FKNN) is a classification algorithm which overcomes the limitations of the KNN classifier by incorporating the fuzzy set theory into KNN, FKNN assign fuzzy memberships of sample to different classes. The class assigned to be the winner is the one which has the highest membership degree.

For pattern x , the FKNN algorithm attributes a membership vector as a function of the pattern's distance from its k-nearest Neighbours, this guarantees that no arbitrary assignments are made.

The first step to be carried out for the FKNN algorithm is computing the fuzzy partition matrix $U = (u_{ij})$ from the memory where a set of n training sample vectors $(x_1, \dots, x_n)$ are stored. In this work, j is represented as the vector index ($j = 1, 2, \dots, n$), where n is the number of training samples, the variable i represents the class index ($i = 1, 2, \dots, C$), and C is the number of classes. For each training case x , we identify its k nearest Neighbours by computing Euclidean distances. The membership degree of the sample vector x_j in the class i is as given in equation 1.

$$u_{ij}(x) = u_i(x_j) = \begin{cases} 0.51 + (n_i / k) \times 0.49, & \text{if } c(x_j) = i \\ (n_i / k) \times 0.49, & \text{if } c(x_j) \neq i \end{cases} \quad (8)$$

Where n_i is the number of Neighbours discovered to belong to the class i and $c(x_j)$ represents the class label of the sample vector x_j . n_i represents labelled reference patterns amongst the k closest labelled reference patterns which are labelled in class x_i , and j ranges from 1 to n . The memberships are shared among the classes. It can be clearly seen that u_{ij} is an element of the C -by- n matrix U .

Equation 8 allocate higher fuzzy membership values to the training samples that are far away from the decision boundary and lower fuzzy membership values to the patterns which lie in the neighbourhood of the decision boundary, this is because the information supplied by the samples in the region close to the decision surface is more uncertain than those provided by other samples. Since u_{ij} is a fuzzy membership grade or value of the sample x_j in the class i , u_{ij} must satisfy the following properties:

$$u_{ij} \in [0,1] \quad (9)$$

$$\sum_{i=1}^C u_{ij} = 1 \quad (10)$$

Where $0 < \sum_{j=1}^n u_{ij} < n$

The second step of the FKNN algorithm is to assign fuzzy memberships of the unknown sample x to different classes according to the following equation:

$$u_i(x) = \frac{\sum_{j=1}^k u_{ij} (1 / \|x - x_j\|^{2/(m-1)})}{\sum_{j=1}^k (1 / \|x - x_j\|^{2/(m-1)})} \quad (11)$$

$$C(x) = \arg \max_{i=1}^C (u_i(x)) \quad (12)$$

where $i = 1, 2, \dots, C$, and $j = 1, 2, \dots, k$. In this context, j represents the j^{th} sample vector among the k nearest neighbors of x . C is the number of classes; k denotes the Neighbouring size. The fuzzy strength parameter m is used to ascertain how heavily the distance is weighted when calculating each Neighbour's contribution to the membership value and is determined *a-priori* from the training data. The value of m is usually taken as $m \in (1, \infty)$. $\|x - x_j\|$ denotes the distance between x and its j^{th} nearest Neighbour x_j . In this research, Euclidean metric is used as the distance measurement, and, u_{ij} , which denotes the membership degree of the sample vector x_j in the class i , is computed in the first step of the algorithm. [14].

IV. Methodology

This section discusses the steps, as shown in figure 1, or workflow of the intrusion detection system which includes data acquisition, feature extraction, data normalization, classification and evaluation.

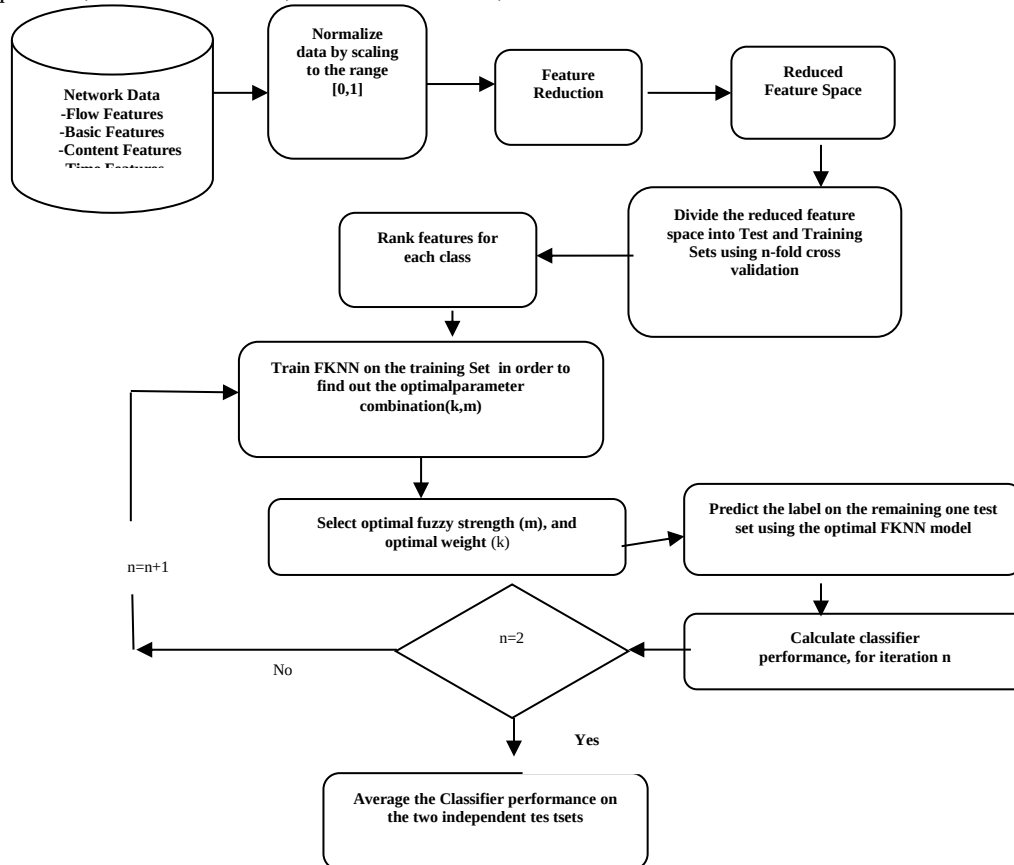


Figure 1: The overall procedure of the FKNN-based model

Data Acquisition In deploying methods for intrusion detection in this work, the data used was obtained online at Source: <https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/>. [15]. The dataset consists of 300 data cases, 250 normal, and 50 intrusive user activities.

Feature reduction This is done by applying principal component analysis on the dataset to extract relevant dataset.

Classification phase The FKNN classifier is founded on the approximation of the membership functions for the labelled reference patterns. Techniques for automatic estimation of membership functions were summarized in the works of [14]. There are two types of labelling namely: fuzzy labelling, otherwise known as soft labelling and crisp labelling.

This paper is based on the fuzzy labelling methods, where $u_i(x)$ is the membership of the test vector x , to class i , $\|x-x_j\|$ is the Euclidean distance between the test vector x , and the k -th nearest Neighbour vector x_j , and m is a real number greater than or equal to 1.0 that sets the “strength” of the fuzzy distance function. An interval of (1, 2) was chosen and continually increased with steps of 0.01 for the fuzzy strength parameter m , and subsequently classification performance was confirmed by way of the 2-fold CV analysis on various values of nearest Neighbours k . For every value of m in the interval (1, 2), the average accuracy obtained by FKNN by the way of Cross Validation (CV) analysis, and eventually, the one with the highest average accuracy was selected as the optimal fuzzy strength parameter. After choosing the optimal fuzzy strength parameter, the FKNN classifier was employed to compute the classification accuracy using the reduced feature set, and then an average of the results was taken. The pseudo-code for the classification phase is shown in figure 2.

Figure 2: Pseudo-code for the classification procedure

```

/* Estimation by using n-fold CV where n = 2*/
Begin
For i = 1:Mmax
For j = 1:k
Training set = k - 1 subsets;
Test set = remaining subset;
Train the FKNN model on the training set to find the
optimal fuzzy strength parameter m when the
Neighborhood size k is set to 3, 5, 7, 8 and 9
respectively;
Test it on the test set and assigns the accuracy to V(j),
where V is a vector whose element is the
corresponding accuracy obtained by each folder;
End for
Compute the mean value of vector V, and store the
mean CV accuracy in the vector M (i);
End for
Get the optimal m value whose corresponding mean CV
accuracy is the highest in M (i);
End
Begin
For l = 1: k
Training set = k - 1 subsets;
Test set = remaining subset;
Train the FKNN model on the training set using the
obtained optimal parameter combination;
Test it on the test set and save the mean CV accuracy;
End for
Return the average classification accuracy rates of FKNN over l test set.
End.

```

V. Results And Discussion

The model simulation was carried out in MATLAB environment using MATLAB version R2015a (8.1.0.604). The proposed system implementation interface is shown in figure 3. Figure 4 shows sample screen of the Principal Components projected on to the first dimensions. The system used 150 data cases for model construction (training) and 150 data cases for testing, this implies that 50% of the data is set aside for model training while the other 50% is use for testing the model. The whole dataset was divided into two data folds where each fold in turn was used for testing. The model performance was evaluated by way of average.

Figure 3: FKNN model Interface

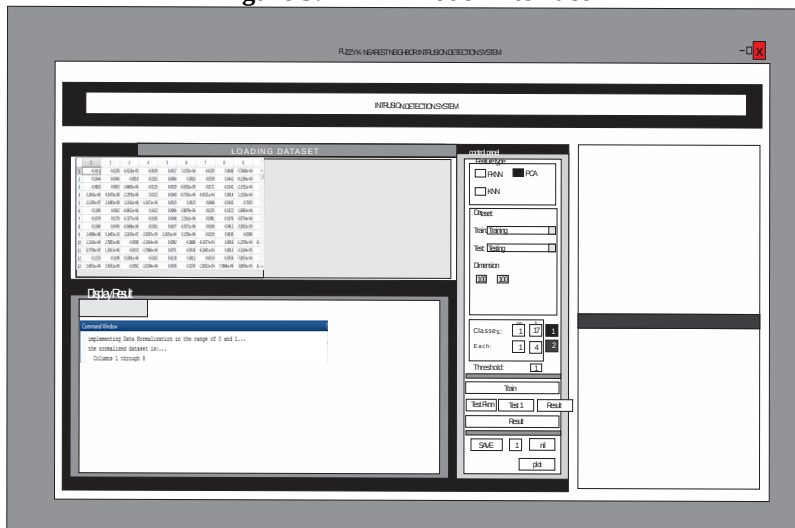
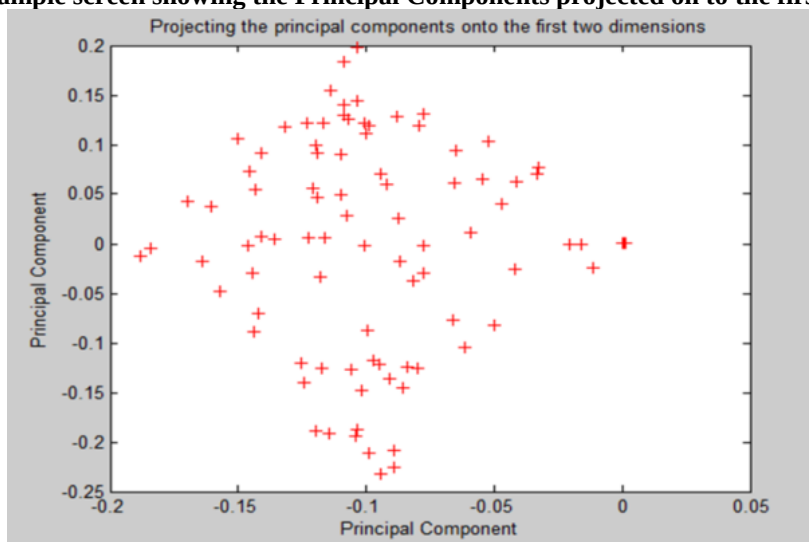


Figure 4: Sample screen showing the Principal Components projected on to the first dimensions



The model performance was evaluated by way of average predictive results of the two folds. The process described above is known as two-fold cross validation. The results as shown in table 1, showed that the system produced averages sensitivity, specificity, accuracy of 85.6%, 72.0%, 92.3% and 88.2% respectively for FKNN with PCA, while it produced 60.92%, 65.11% and 64.51% for sensitivity, specificity, accuracy respectively for FKNN without PCA. These results are better than that of KNN with PCA of 62.2%, 70.0% and 76.0% for sensitivity, specificity, accuracy respectively and sensitivity of 47.2%, specificity of 50.1%, and accuracy of 55.2% for KNN without PCA.

Table 1: Results of evaluations of FKNN and KNN with Metrics

Algorithm	Sensitivity (%)	Specificity (%)	Accuracy (%)
KNN without PCA	47.20	50.11	55.20
KNN with PCA	62.20	70.00	76.00
FKNN without PCA	60.92	65.11	64.51
FKNN with PCA	88.20	92.25	85.67

The graphs of ROC Classifier performance for fold 1 and fold 2 at different set parameters are shown in figure 5, figure 6, figure 7 and figure 8.

Figure 5: ROC Classifier performance for fold 1, when m=1.88 and k=7

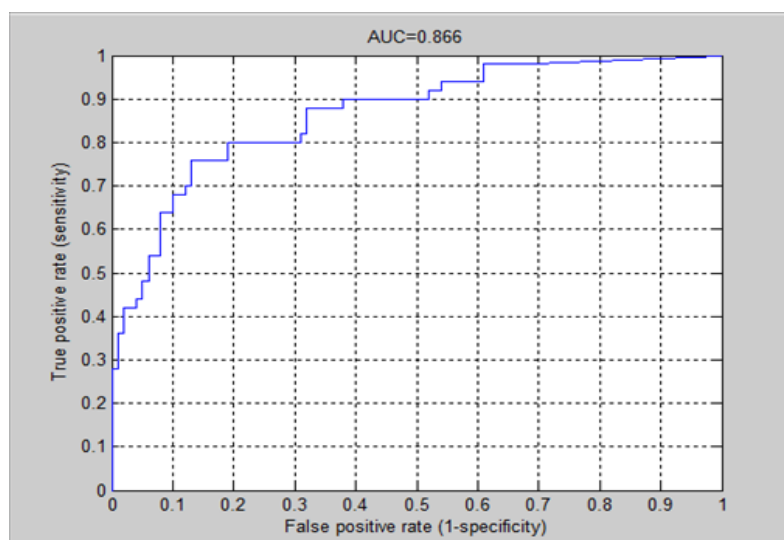


Figure 6: ROC Classifier performance for fold 2, when $m=1.88$ and $k=7$

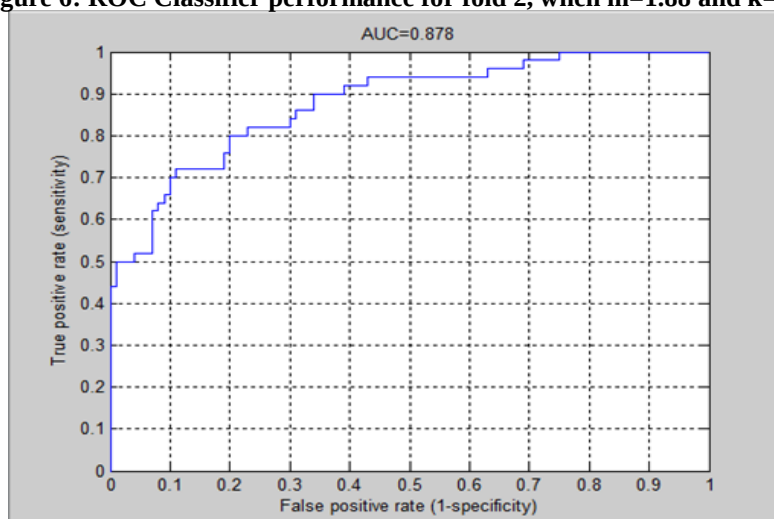


Figure 7: ROC Classifier performance for fold 1, when $m=2.00$ and $k=9.00$

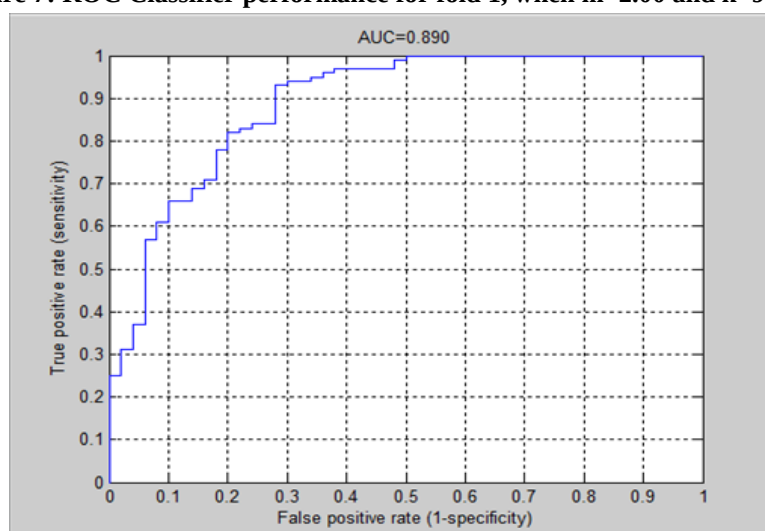
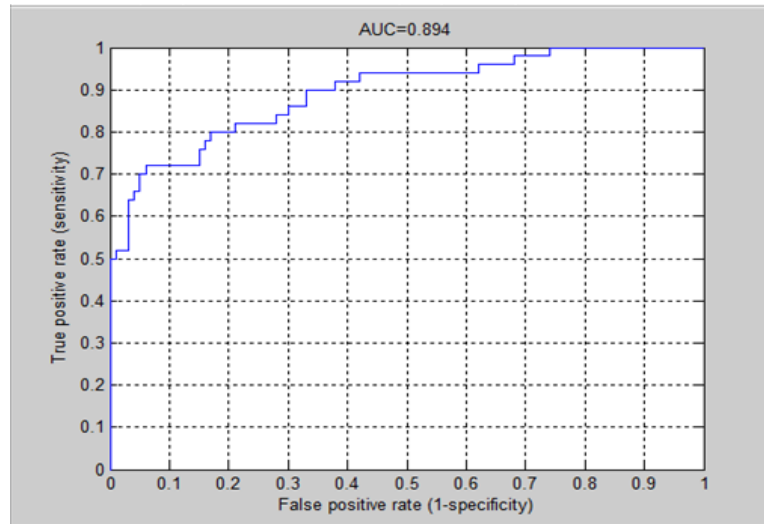


Figure 8: ROC Classifier performance for fold 2, when $m=2$ and $k=9$



VI. Conclusion

IDS modelling are critical tasks and in recent years, various techniques of data mining and soft computing are being explored to strengthen network security which includes neural network, decision tree, genetic algorithm, support vector machine, Naïve Bayes classification, clustering, fuzzy logic, etc. Wireless communication between mobile users is becoming more popular, which is due to recent technological advancement in mobile computers and wireless communication devices such as wireless modem and Wireless LAN. There are various security requirements to be handled in wireless network such as confidentiality, availability, integrity. The area of research that can aid this type of potential threat is anomaly detection. Thus, analysis and implementation of the developed fuzzy k-nearest Neighbour-based model used in detecting intrusive network user activities have been presented. And the results showed that FKNN will improve on the loss of revenue faced by the telecommunication industries.

VII. References

- [1]. Offurum, S.C., (2009). Communication Revolution, The Leader, vol.1, issue 19, Pp 2.
- [2]. Thomas, J., Jain, R., Abraham, A., and Han, S.Y., (1997). D-SCIDS: Distributed Soft Computing Intrusion Detection System, Journal Of Network And Computer Applications, Pp 1-19.
- [3]. Ko, C., Fink, G., and Levitt, K., (1994). Automated detection of vulnerabilities in privileged programs by execution monitoring. In Proceedings of the 10th Annual Computer Security Applications Conference, vol. 10, Pp 134-144.
- [4]. Kosoresow, A.P., and Hofmeyr, S.A., (1997). Intrusion Detection via System Call Traces. IEEE Software, Pp 35-42.
- [5]. Forrest, S., Hofmeyr, S.A., Somayaji, A., and Longstaff, T.A., (1996). A Sense of Self for Unix Processes. in Proceedings IEEE Symposium on Security and Privacy, IEEE Computer Society Press, vol. 11, Pp 1-9.
- [6]. Axelsson, S., (2000). Intrusion Detection Systems: A Survey and Taxonomy, Pp 1-27.
- [7]. Liao, Y., and Vemuri, V.R., (2002). Use of K-Nearest Neighbor Classifier For Intrusion Detection, Vol. 21, issue 5, Pp 439-448.
- [8]. Chitrakar, R., and Huang, C., (2012). Anomaly Detection using Support Vector Machine Classification with K-Medoids Clustering, Pp. 1-5.
- [9]. Lee, W., Stolfo, S., Eskin, E., Miller, M., Zhong, Z., and Yi, G., (2002). Adaptive Model Generation for Intrusion Detection Systems. Pp 1-14.
- [10]. Zhou, M., and Huang H, Wang Q., (2012). A Graph-based Clustering Algorithm for Anomaly Intrusion Detection. Pp 1311-1314.
- [11]. Jianliang, M., Haikun, S., and Ling, B., (2009). The Application on Intrusion Detection based on K-Means Cluster Algorithm. In Proceedings of International forum on Information Technology and Application, IFITA, Vol. 1, Pp 150-152.
- [12]. Ibrahim, J., (2002). 4G Features. International Journal of Engineering Science and Innovative Technology, (IJESIT), Vol. 2, Issue 4, Pp 1-4.
- [13]. Muda, Z., Yassin, W., Sulaiman M.N., and Udzir, N.I., (2011). Intrusion Detection based on K-Means networks and support vector machines. International Journal of Network Security & Its Applications (IJNSA), vol.5, No.1, Pp 648- 655.
- [14]. Keller, J. M., Gray, M. R., and Givens, J. A., (1985). A Fuzzy K-Nearest Neighbours. IEEE Transactions On Systems, Man, and Cybernetics, Vol.15, No. 4, Pp 1-6.
- [15]. Nour, M., and Jill, S., (2015). The Significant Features of the UNSW-NB15 and The KDD99 Data Sets For Network Intrusion Detection Systems. Vol.5, Pp 1-8. <https://www.researchgate.net/publication/299298212>.
- [16]. Anderson J.P. (1980). Computer security threat monitoring and surveillance, Technical report, James P Anderson co., Fort Washington, Pennsylvania.
- [17]. Denning, D.E. "An Intrusion-Detection Model", IEEE Transactions on Software Engineering, vol. 13, no., pp. 222-232, February 1987, doi:10.1109/TSE.1987.232894
- [18]. Jolliffe, I. (1986). Principal Component Analysis. Harrisonburg: R. R. Donnelley & Sons.
- [19]. K. M. Leung (2007). k-Nearest Neighbor Algorithm for Classification, POLYTECHNIC UNIVERSITY, Department of Computer Science / Finance and Risk Engineering, mleung@poly.edu
- [20]. P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min and W. Worek, "Overview of the Face Recognition Grand Challenge," in Computer vision and pattern recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 2005, pp. 947-954.