https://afrjcict.net

An Intrusion Detection System using Support Vector Machine and Infinite Latent Feature Selection Approach

B. C. Asiegbu[1], C. O. Ikerionwu[2*] and N. C. Ajanwachuku[3]
Department of Information Management Technology,
Federal University of Technology, Owerri,
Nigeria
Email: [1]cbasiegbu@yahoo.com,
[2]charles.ikerionwu@futo.edu.ng,
[3]ajanwachukunwagu@gmail.com
**Corresponding author*

ABSTRACT

For many decades, network security has faced serious threats from unauthorized users (eavesdroppers, network sniffers, social engineers and the evil players also known as the network hackers) and as such, government institutions, financial institutions, and academic institutions have lost relevant organisational information that is worth billions of naira to individuals and organisations that are not entitled to this information. Machine learning systems for network security, popularly known as intrusion detection systems (IDS), can help to minimize the menace caused by the evil players. This paper showed how to use an effective machine learning algorithm known as support vector machine (SVM) model and an effective and efficient data engineering method (Infinite latent feature selection) to develop an enhanced IDS for network security. The implementation was simulated using LibSVM 3.22 in a MATLAB integrated development environment and the KDD Cup' [99] dataset- a standard evaluation benchmark for intrusion detection systems was used to train and test the effectiveness and efficiency of the IDS. The performance indicators used to measure the effectiveness of the proposed system is detection rate and false alarm rate. The enhanced IDS achieved a detection accuracy of 99.88%, and 97.91% for normal network packets and attack network packets respectively, which is an improvement over the existing rate of 64.94% and 70% for detection accuracy of normal network packets and attack network packets respectively. By implementing the enhanced IDS, in addition to efficient data engineering, network security experts could improve the accuracy of data classification, intrusion detection rate and low false alarm rate.

Keywords: *Detection, Support Vector Machine, Network, Dataset, Infinite latent feature selection.*

African Journal of Computing & ICT Reference Format:

B. C. Asiegbu, C. O. Ikerionwu and N. C. Ajanwachuku (2019),
An Intrusion Detection System using Support Vector Machine
and Infinite Latent Feature Selection Approach,
Afr. J. Comp. & ICT, Vol.12, No. 3, pp. 74 - 84.

I. INTRODUCTION

The United States Department of Justice officially charged nine Iranian evil players (hackers) over series of cyber-crimes on more than three hundred universities in the United States and abroad. The hackers looted intellectual resources worth three billion USD [1]. This cyber-crime is one of the worst ever recorded in 2018. The evil players have not for one day rested in their quest to infiltrate information systems and computer networks belonging to governments, public and private organizations. Why then should the good players (Cyber security experts) rest in the fight to combat the evil players from compromising relevant information systems and computer networks? We need to find a best way of protecting these systems. An intrusion detection is an unauthorized access to relevant information stored in information systems with the sole objective of using this information for callous reasons [2]. Different security mechanisms have been used in the past to prevent unauthorized access to relevant information systems. For example, the use of authentication security platforms (password based authentication, token based authentication, and biometric based authentication).

Also, the traditional approaches (firewalls, spyware, antivirus etc.) have been used too. These approaches, no doubt have tried to an extent to prevent the activities of the evil players. However, they have loopholes which are now known and exploited by the evil players. They (evil players) use these loopholes to compromise the integrity of information systems and computer networks. Further, [2] described intrusion detection system as a security mechanism designed to detect attacks among the various types of network packets. It is used to examine events that occur in information systems or computer networks and analyzing them for presence of intrusion. The intrusion attacks are of two categories: the network based and the host attacks. In the network based attacks, the attacker takes over the network and its resources, and prevents authorized users from accessing various network services. While in the host network based attacks, the attacker targets the information system and tries to gain unauthorized access to privileged services or resources on that information system [6].

Intrusion detection systems are of two types; the misuse detection and the anomaly detection systems. Misuse detection systems identify unauthorized access based on confirmed pattern of malicious activities [4]. The anomaly detection systems try to identify malicious traffic based on deviation from the known network traffic signatures [3]. Therefore, this paper aims to demonstrate an improved IDS for network based attacks using a supervised machine learning algorithm (SVM) for dataset classification and infinite latent feature selection for dataset engineering. This paper is organized as follows: section two presents a review of related works, SVM algorithm and infinite latent feature selection. Section three describes the proposed system, its architecture, implementation and test. Section four presents the experimental result and discussion, and finally, section five concludes the work.

II. LITERATURE REVIEW

2.1 Review of Empirical framework

[5] demonstrated the application of fuzzy logic in reducing the false alarm rate while determining intrusive activities. The work defined a set of fuzzy rules which are used in defining the normal and abnormal behavior in a computer network, and a fuzzy inference engine which is used to determine intrusions. The work also used a genetic algorithm (GA) to generate fuzzy classifiers, which is a set of fuzzy rules as defined already. Each fuzzy rule is represented by a genome and the GA is used to find the best genomes (fuzzy rules) to be added to the fuzzy classifier. The authors conducted experiments using the KDD evaluation data to classify 22 different types of attacks into 4 intrusion classes: denial of service (DoS), unauthorized access from a remote machine (R2L), unauthorized access to local super user (root) privileges (U2R), and probing (PRB). The results showed that the algorithm of the study achieved an overall true positive rate of 98.95% and a false positive rate of 7%. [7] used self-organizing map (SOM) and artificial neural network (ANN) machine learning algorithms in designing IDS.

The machine learning system of the study has an objective of addressing the problem of anomaly and misuse detection in a computer network. The key performance indicator for measuring the effectiveness of the system developed in this study are detection rate (DR), false alarm (FA), and false positive (FP). The implementation of the system developed in this study was tested using the 1998 DARPA dataset. A detection prediction of 97.1% and a false positive of 2.8% were achieved in this study. In another related work, [8] used a hybrid classifier that consists of genetic algorithm (GA), artificial neural network (ANN), K-NN, and support vector machine (SVM) to design a machine learning system for network

security. This is meant to address the issue of anomaly detection. The key performance indicators for evaluating the system that was developed are detection rate (DR), false positive (FP), and false negative (FN). The implementation of the system developed in this study was tested using the 1998 DARPA dataset. A detection prediction of 98.6%, false positive (FP) of 2.5% and a false negative of 11% were achieved in this study. [6] demonstrated the need for a carefully constructed training and test corpora, effective feature extraction and selection, and a valid evaluation on representative corpora when applying pattern classification research.

This new approach is aimed at providing sustained good performance in adversarial environments where a malicious adversary takes actions to subvert a classifier and this objective was achieved. [9] analyzed executable instead of email messages. The study demonstrated that N-gram analysis of executables can be used to distinguish between normal computer programs and malicious virus, worm, Trojan horse programs. The study achieved a detection prediction accuracy of 98% for 291 previously unseen malicious executables at a false alarm rate of 5%, an impressive success indeed. These good results were achieved by collecting and carefully confirming and labeling a training corpus of 1,971 mild and 1,651 malicious executables and using 10- fold cross-validation to select both the top performing N-grams and the best performing classifier (a boosted tree classifier). However, it should be noted that approximately 20%ofthe malicious software samples used were concealed with either compression or encryption. [6] suggest an enhanced training time to support vector machine (SVM), especially when dealing with large datasets, and hierarchical clustering analysis.

The study used the Dynamically Growing Self Organizing Tree (DGSOT) algorithm for clustering. This algorithm (i.e., DGSOT) used in this study overcame the bottleneck posed by the traditional hierarchical clustering algorithm (e.g. hierarchical agglomerative clustering). The approach used was very successful in terms of detection accuracy, false positive and false negative reduction. [10] used a new technique (K – Means neural network approach) which combined a multi layered neural network algorithm and a back- propagation feed forward learning algorithm to classify and detect network anomalies from a collected dataset of network traffic. After carrying out the experiment, the proposed system of the study achieved a detection success of 92% and a false alarm rate of 6.21%. [12] used a single classifier i.e., a Self-Organizing Map (SOM) algorithm to design a machine learning system that addressed the problem of anomaly detection in the network. The key performance indicators for evaluating the machine learning system developed are false positive (FP), and detection rate (DR). The implementation of the system developed was tested using the KDD-Cup ’99 dataset. The study achieved a detection prediction rate of 90.4%, a commendable success and a false positive (FP) of 1.38%. [11] used genetic fuzzy classifier algorithm to design a machine learning system for network security. The study addressed problem of anomaly and misuse detection that is common in network security as its objective. The researchers adopted detection rate (DR) as a key performance indicator while measuring the effectiveness of the developed system. The implementation of the system developed was tested using the KDD-Cup ’99 dataset and a detection rate of 53.02% was achieved. [13] used a hybrid classifier which consists of decision tree (DT) and support vector machine (SVM) algorithms to design a machine learning system in addressing the problem of anomaly detection in the network.

The key performance indicators used to evaluate the effectiveness of system developed are accuracy (AC), false positive (FP), and detection rate (DR). The implementation of the system was tested using the KDD-Cup ’99 dataset. The system recorded a huge success in false alarm reduction of 1%. However, the detection rate (70%) and the accuracy (64.94%) recorded by the system was below average and is open for further improvement.

2.2 Support Vector Machine Analysis

SVM is a supervised learning system that maps linear algorithms into non-linear spaces. It is based on the idea of a hyper-plane (classifier) or linearly separability [6]. How does this algorithm work? To explain this, we would adopt the quadratic equation presented by [14].

Let “N”, which is a natural number have the following data points:

$\{(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), (X_4, Y_4),..., (X_n, Y_n)\}$, where $X_i \in R_d$ (X_i is a training vector which is an element in a set of real numbers that are data points, R_d) and $Y_i = \{+1,-1\}$, i = 1, 2, ….n.

The classifier is defined by the weight vector (w), and the bias (b). Where a new feature vector “x” can be classified with the following function:

$F(x) = sign(w. x + b) = sign(K (X_1. X) + b)$……….1

In equation 1, where "K" is the kernel function, the training vectors X_1 occurs only in the form of a dot product with each training point having a Lagrangian multiplier α_i.

The value of the Lagrangian multiplier shows the importance of each data point. Vectors that are found closest to the classifier have $\alpha_i > 0$ and are called support vectors. Hence, all other vectors have $\alpha_i = 0$. The support vectors represent the classifier and show how reliable the classifier is. Fig 1 highlights how the SVM algorithm works on captured dataset. Data with the same characteristics are grouped into the same class. This classification is based on equation (1).

2.3 Infinite Latent Feature Selection Analysis

The objective of infinite latent feature selection (ILFS) is to provide distinguishing importance for each feature belonging to a large dataset. For instance,
if $\boldsymbol{\beta} = \{V0 = i, V_1, \ldots, V_i - 1, Vi = j\}$ equals a path of length L between node i and j, that is, $\vec{X_i}$ and $\vec{X_j}$, through other features $V_1, \ldots, V_i - 1$ and the length L of the path is lower than the total number of nodes n in the graph. In this context, a path is simply a subset of the available features or nodes that come into play. However, the network is characterized by walk structure, where nodes and edges can be visited multiple times [15]. The joint probability that β is a reliable subset of the feature is given by:

$$\rho_\beta = \prod_{K=0}^{L-1} a_{vk,vk+1} \quad \ldots(2)$$

$P_{i,j}^L$ equals all the paths of length L between i and j; it is summed up as follows:

$$C_L(i, j) = \sum_{\beta \in P_{i,j}^L} \rho\beta, \quad \ldots (3)$$

When expressed in standard matrix algebra it is given as C_L (i, j) = A^L(i, j), meaning that the adjacency matrix A raised to the power L, where L is length of path between nodes or features. Considering all possible paths of any length means extending the path length to infinity and it is thus calculated as the geometric series of matrix A

$$\hat{C} = \sum_{L=1}^{\infty} A^L \quad (4)$$

Summing infinite A^l terms brings divergence [17]. There is a need for regularization at this point. That is:

$$\check{C} = \sum_{L-1}^{\infty} r^l \; A^l \quad \ldots(5)$$

Where r^l= generating function for the l – path. Computing (5) using the convergence property of the geometric power series of a matrix [16]. we get

$$\check{C} = (I - rA)^{-1} - I \quad \ldots(6)$$

$\check{C}$ now encapsulate all the information about the correctness of our set of features. At this stage, we obtain final importance for feature by marginalizing. That is,

$$\check{C}(i) = [\check{C}\mathbf{e}] \quad (7)$$

where **e** stands for a 1D array of ones [17]. The output of the ILFS algorithm is ranked in a decreasing order, in this case, scores, it implies that the output of the ILFS algorithm is, thus, a ranked list of relevant features.

III. SYSTEM ARCHITECTURE AND IMPLEMENTATION

3.1 System Architecture

The system has a graphical user interface (GUI) for the collection of KDD Cup"99dataset. After data collection, the dataset undergoes data engineering. The pre-processing involves transforming the datasets into the format it can be used on Matlab (this transformation is done using C# programming language), and feature selection and reduction (this involves selecting relevant dataset features and discarding irrelevant and redundant dataset features) using Infinite latent feature selection method. Next, the SVM model is trained using relevant Dataset obtained from the feature extraction process. The SVM model is tested after the training phase has been completed in evaluating the learning capability of the IDS. The machine learning system is said to have learnt well if its accuracy in classification and detection rate are high and if its false alarm rate is low. Fig 2 shows the system architecture.

3.2 Implementation and Test

The proposed system was implemented using MATLAB and C#. It consists of three major modules, which are:
Load, Training and Testing/Classification modules.

Load Module: This module loads the standard Dataset (KDD Cup"99) for training. .
The KDD Cup"99dataset (10,000 data points) is entered into the system for pre-processing and training through the load graphical user interface (GUI) as shown in Fig 3

After the KDD Cup"99dataset has been loaded into the system, the pre-processing operation (feature selection and reduction) starts. The KDD Cup"99dataset has 41 data features. However, after the pre-processing, operations are carried out by the system's feature selection algorithm (infinite latent feature selection), where 23 top ranked features were selected as shown in figure 4.

Next, the IDS is trained with the 23 top ranked features that were selected

Training Module: This module enables the training of the IDS with the 23 top ranked features from the standard dataset (KDD Cup"99). Figure 5 shows the successfully trained IDS with the 23 top ranked features.

Testing / Classification Module: This module is used to test the machine learning system for network security to verify how much it has learnt. This module classifies network packets into two classes (normal and attack). Abnormal network packets are classified under the attack class, and trusted network packets are classified under the normal class. How do we ascertain how much the IDS have learnt? We do that by taking note of the number of misclassified packets. If the number of misclassified packets is much, then the machine learning system didn't learn much and vice versa. The proposed system was trained with 2,000 data points and it did achieve a misclassification rate of 0.005, which is an improvement. The result of the proposed system is shown in figure 6.

IV. RESULTS AND DISCUSSION

4.1 Experimental Results Presentation

The results from the experiments carried out are presented in Table 1.

An IDS is said to have achieved a detection accuracy of 100% if it records no misclassification of data. Calculating the different percentage of misclassification for normal and attack network packets and subtracting it from 100% gave us our system's detection accuracy (DA) for each class. The DA for normal network packets is derived as follows:

$$\text{Detection Accuracy (DA)} = \frac{FP}{FP+TN} \times 100 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

where FP = False Positive, TN = True Negative,
Therefore, substituting the values
in (2), we get:

$$\frac{2}{2+1616} \text{ x } 100 = 0.12$$
$$100 - 0.12$$
$$= 99.88$$

To calculate the detection accuracy for attack network packets, we adopt a similar formula:

$$\text{Detection Accuracy (DA)} = \frac{FN}{FN+TP} \times 100 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

where FN = False Negative, TP = True Positive

Substituting the value in (3), we get:

$$\frac{8}{8+374} \text{ x } 100$$

$$= 2.09$$
$$= 100 - 2.09$$
$$=97.91$$

4.2 Discussion

In this paper, 10,000 data points were used to train the proposed intrusion detection system, and 2,000 data points were used to test its efficiency. The experimental result shown in figure 6 shows that 1,616 datasets were classified as normal network packets and 374 network

packets were classified as attack network packets while a very few datasets were misclassified. This is an indication of reduced false alarm rate and improved detection accuracy (performance indicators which the proposed intrusion detection system was measured with). We achieved a detection accuracy of 99.88% and 97.91% for normal network packets and attack network packets respectively. [13] used a hybrid classifier (SVM and decision tree) for the IDS developed in the study and achieved a success rate of 64.94% and 70% for detection accuracy of normal network packets and attack network packets respectively as against our success rate of 99.88% and 97.91%. In this paper, we have successfully demonstrated an improved intrusion detection system for computer network security using a supervised machine learning algorithm (SVM) for classification of network packets and a statistical approach (Infinite latent feature selection) for effective and efficient data engineering.

V. CONCLUSION

In summation, this work sought to enhance and intrusion detection system using a support vector machine (SVM), and an infinite latent feature selection (ILFS) approach. The enhanced IDS achieved a detection accuracy of 99.88%, and 97.91% for normal network packets and attack network packets respectively, which is an improvement over the existing rate of 64.94% and 70% for detection accuracy of normal network packets and attack network packets respectively. By implementing the enhanced IDS, in addition to efficient data engineering, network security experts could improve the accuracy of data classification, intrusion detection rate and low false alarm rate which are key performance indicators for evaluating machine learning systems for network security.

REFERENCES

[1] H. N. Lily, "The worst cyber security breaches of 2018 so far," September 7 2018. Available at https://www.wired.com/story/2018-worst-hacks-so-far/

[2] A. Kalekar, N. Kshatriya, S. Chakranarayan, and S. Wadekar, "Real Time Intrusion Detection System using Machine Learning", "*In International Journal of Engineering Research and Technology*", Vol. 3, No. 2, Feb 2014.

[3] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection: Support vector machines and neural networks," "*In: Proceedings of the IEEE international joint conference of neural networks (ANNIE)"*, pp. 1702 – 1707, St. Louis, Mo, 2002.

[4] A. Chauhan, G. Mishra, and G. Kumar, "Survey on data mining techniques in intrusion detection". "*International Journal of Scientific & Engineering Research"*, Vol. *2*(7), pp. 1-4, 2011

[5] J. Gomez, and D. Dasgupta, "Evolving fuzzy classifiers for intrusion detection", "*In Proceedings of the 2002 IEEE Workshop on Information Assurance*", West Point, NY, USA, 2002.

[6]K. Latifur, A. Mamoun and T. Bhavani, "A New Intrusion Detection System Using Support Vector Machines And Hierarchical Clustering", "*The VLDB Journal"*, Vol. 16, pp.507-521, 2007.

[7] Y. Liu, K. Chen, X. Liao and W.Zhang "A genetic clustering method for intrusion detection, and Pattern Recognition," "*Pattern Recognition Society*", Vol. 37, pp.927– 942, 2004.

[8] T. Shon, J. Moon, A hybrid machine learning approach to network anomaly detection, Information Sciences, 177,18:3799-3821 (2007).

[9] J. Kolter, and M. Maloof, " Learning to detect and classify malicious executables in the wild, "*Journal of Machine Learning Research",* 7:2721–2744, 2006.

[10] K. M. Faraoun and Boukelif, " Neural Network Learning Improvement Using the K – Means Clustering Algorithm to Detect Network Intrusions",

"*International Journal of Computer and Information Engineering*", Vol. 10, 2007.

[11] T. Ozyer, R. Alhajj, and K. Barker, "Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening," "*Journal of Network and Computer Applications,"* 30 pp.99 – 113, 2007.

[12] H.G. Kayacik, Z.H. Nur, and M.I Heywood, "A hierarchical SOM-based intrusion detection system", "*Engineering Applications of Artificial Intelligence*", Vol. 20, pp.439–451, 2007.

[13] W. Su-Yun, and E. Yen, "Data mining-based intrusion detectors," "*Expert System With Application*," Vol. 36, 2009, pp.5605-5612.

[14] A. M. Snehal, P. R Devale, and G. V. Garje, "Intrusion Detection System Using Support Vector Machine and Decision Tree," *"International Journal of Computer Applications," Vol.* 3, No. 3, pp.40-43, 2010.

[15] S. P. Borgatt,i and M. G. Everett, "A Graph-theoretic perspective on centrality," "*Social Networks*," 28(4):466–484, 2006.

[16] H. Hubbard, and B. B. Hubbard, "Vector calculus, Linear Algebra, and Differential forms: A unified Approach (Edition 2)", Pearson, 2001.

[17] R. Giorgio, M. Simone, C. Umberto, and V. Alessandro, "Infinite Latent Feature Selection: A probabilistic Latent Graph – Based Ranking Approach", "*In Conf. IEEE International Conference on Computer Vision*", pages 1398-1406, 2017.

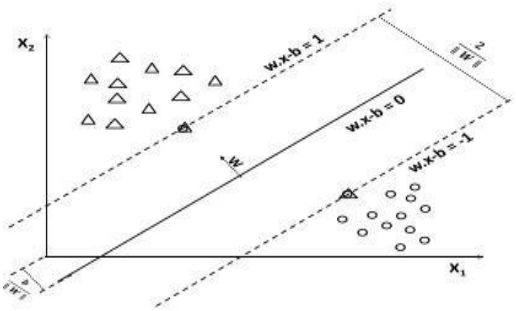

Fig 1: Classification of network packets into two classes (normal and attack) using SVM algorithm.

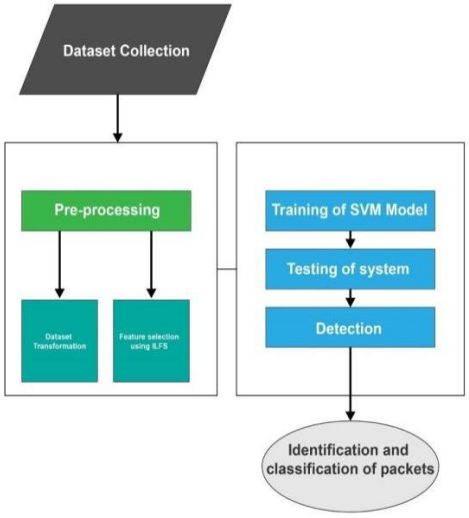

Fig 2: Proposed System Architecture

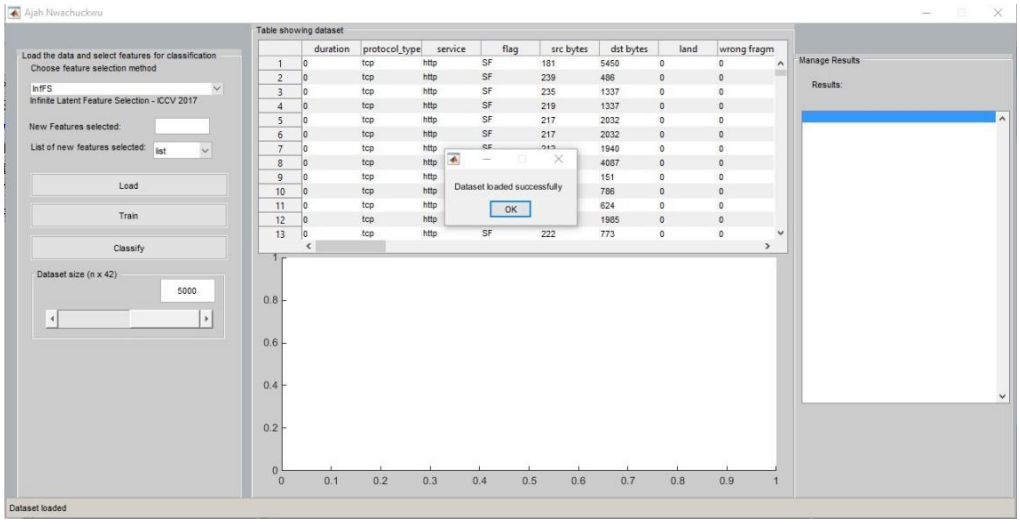

Fig 3: Load module for loading KDD Cup"[99]dataset.

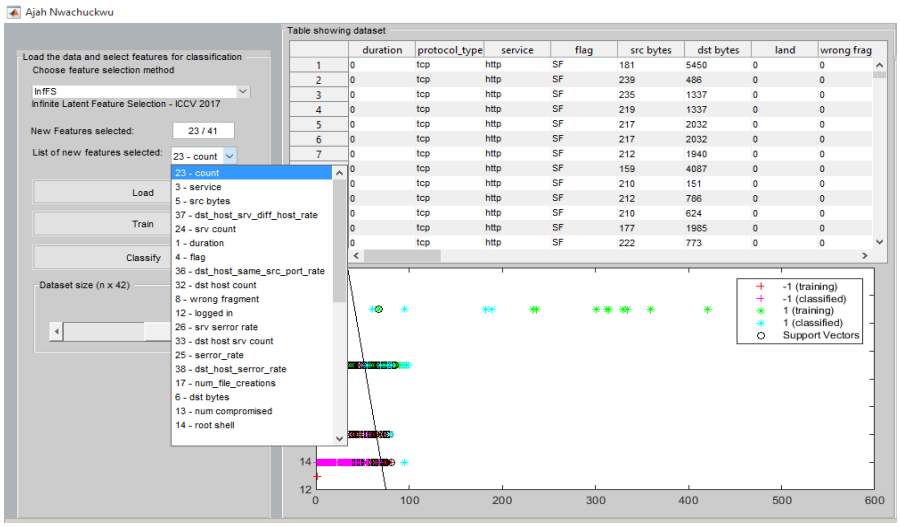

Figure 4: Feature selection

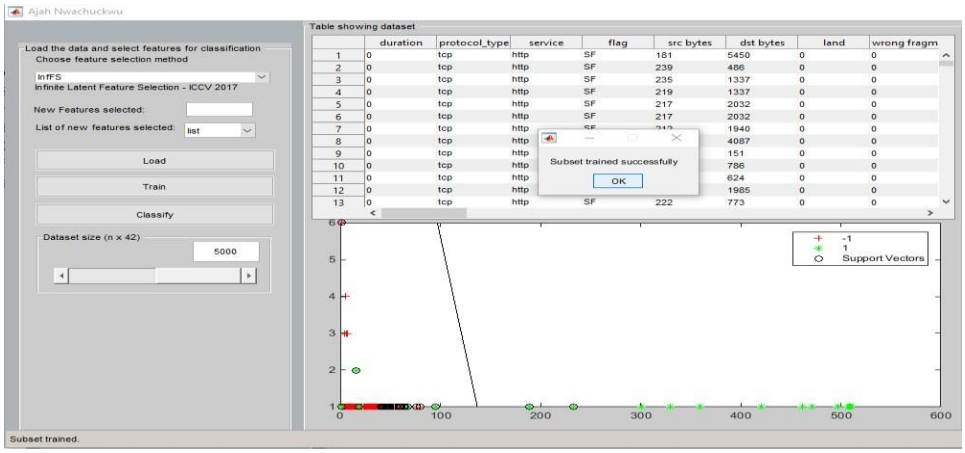

Figure 5: Subsets trained successfully

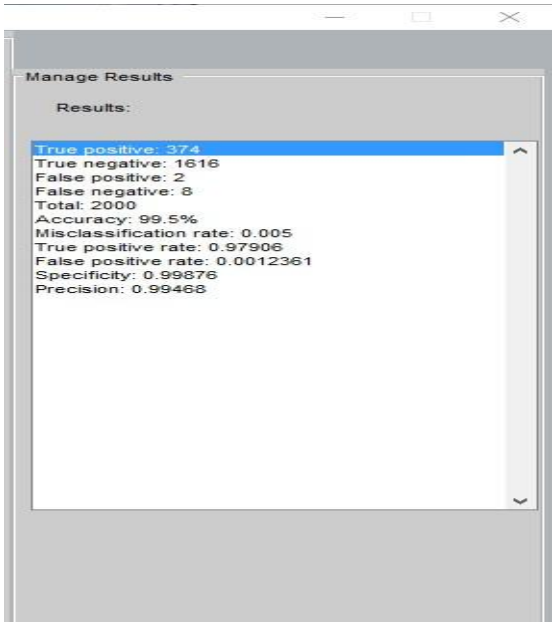

Figure 6: Detection accuracy for normal and attack network packets

Table 1: The experimental results

Class	Normal	Attack
Normal	1616 (TN)	2 (FP)
Attack	8 (FN)	374 (TP)

Table 2: Percentage of normal and attack packets

Class	Accuracy
Normal	99.88%
Attack	97.91%