# Intrusion Representation and Classification using Learning Algorithm

Babu Kaji Baniya
College of Arts and Sciences
Department of Computer Science and
Digital Technologies
Grambling State University
Grambling, Louisiana 71245
Email: baniyab@gram.edu

***Abstract*—At present, machine learning (ML) algorithms are essential components in designing the sophisticated intrusion detection system (IDS). They are building-blocks to enhance cyber threat detection and help in classification at host-level and network-level in a short period. The increasing global connectivity and advancements of network technologies have added unprecedented challenges and opportunities to network security. Malicious attacks impose a huge security threat and warrant scalable solutions to thwart large-scale attacks. These activities encourage researchers to address these imminent threats by analyzing a large volume of the dataset to tackle all possible ranges of attack. In this proposed method, we calculated the fitness value of each feature from the population by using a genetic algorithm (GA) and selected them according to the fitness value. The fitness values are presented in hierarchical order to show the effectiveness of problem decomposition. We implemented Support Vector Machine (SVM) to verify the consistency of the system outcome. The well-known NSL-knowledge discovery in databases (KDD) was used to measure the performance of the system. From the experiments, we achieved a notable classification accuracies using a SVM of the current state of the art intrusion detection.**

***Index Terms*—cybersecurity, intrusion, discriminatory, fitness value, decomposition**

## I. Introduction

Machine learning (ML) algorithms outperform the similar repetitive tasks carried out by security analysts during screening activities. The action can be taken analyzing reports of prior actions by analysts to identify and respond to certain attacks. The models have been trained, and possess enough knowledge to identify a similar attack and respond accordingly without human intervention [1]. Despite this, it is very hard to make a fully automated security system that ultimately replaces human expertise. Therefore, there is a constant need to join task forces (system and human security analysts) to explore network log files analysis, malware detection, and vulnerabilities assessment for network risk analysis. The collective efforts produce robust results and strong defensive mechanisms against a hacker in a network. There are many ML algorithms, such as decision tree and genetic algorithms used to develop applications to create rules for classifying network connection [2]. Other techniques go beyond implementing a cognitive architecture to create an automated cyber defense decision-making system with expert-level ability inspired by human skill [3]. Cybersecurity analysts generally have to spend time responding to multiple events, which sometimes include false positives, which mostly turn out to be a waste of productive time. Therefore, ML classifiers are trained on alert data to identify and separate between false positives and true positives. It will then alert user only on scenarios altered (i.e. true positive) [4].

ML Algorithms are one of the effective techniques to deal with the current cyber-attacks. Algorithms are categorized into three: supervised, semi-supervised, and unsupervised. These algorithms learn the different patterns of normal and malicious activities with a large corpus of both stages (normal, and affected network and host level activities) [5]. In supervised learning, each object consists of an input sample and its corresponding level [6]. This algorithm analyzes the training data and uses the outcomes to map new instances. Unsupervised learning deduces the description of hidden structures from unlabeled data. Because of the lack of ground truth, the accuracy of the algorithm cannot be measured, and only the data pattern can be presented. Semi-supervised learning is the combination of both (supervised and unsupervised) learnings. It uses a limited number of label data to predict the pattern of a large amount of unlabeled data. Therefore, this learning minimizes label efforts while obtaining a high accuracy [7].

An IDS is an efficient detection technique deployed to classify the intrusions automatically at the host and network-level. Based on the attacking behaviors, IDS is categorized into the host-based intrusion detection system (HIDS) and network-based intrusion detection system (NIDS) [1]. An IDS which analyzes characteristics of log files on the user computer in order to detect attacks is called HIDS. An IDS which analyzes network activities is called NIDS [7]. Researchers have already collected a huge amount of datasets (publicly available) for anomaly detection, among them NSL-KDD is one [9]. We introduced the genetic algorithm to find the fitness value (of each intrusion feature), and selected the best fitness values from the intrusion population. It ultimately removed the irrelevant and redundant features from the population, and minimized the computational complexity [12] [14]. In the next stage, the best fitness features are presented into 2-dimensional space for further generalization between normal and malicious

ISBN 979-11-88428-06-9

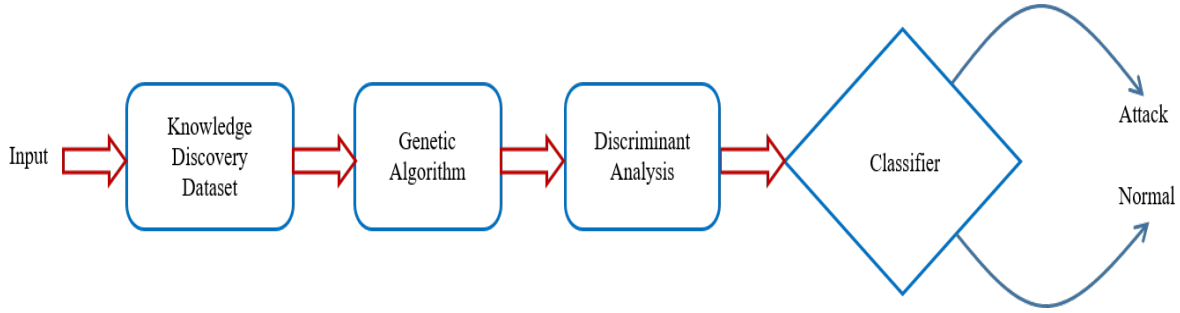



Fig. 1: Overview of IDS: NSL-KDD dataset for evaluation of proposed model, implemented genetic algorithm to find the best fitness value of intrusion population, the best fitness values of each class presented into 2-dimensional space using linear discriminant analysis, and best fitness features feed to the classifier

TABLE I: NSL-KDD dataset and their corresponding classes: normal and attack

| Category | | Description |
|---|---|---|
| Normal | | Normal connection records |
| Attack | DoS | Attacker aims at making network resources down |
| | R2L | Illegal access from remote computer |
| | U2R | Obtaining the root or super-user access on a particular computer |
| | Probe | Obtaining detailed statistics of system and network configuration details |

TABLE II: NSL-KDD dataset attack type and their corresponding sub-categories

| Attack types | Attack term in each category |
|---|---|
| DoS | back, land, neptune, pod, smurf, teardrop, apache2, udpstorm, processtable, worm |
| R2L | guess_passwd, ftp_write, imap, phf, xsnoop, multihop, warezmaster, warezclient, named, spy, xlock, snmpguess, mailbomb, sendmail snmpgetattact, httptunnel |
| U2R | buffer_overflow, loadmodule, rootkit, perl, ps, xterm, sqlattack, |
| Probe | satan, ipsweep, nmap, portsweep, mscan, saint |

intrusions separation boundary.

We further decomposed the problem to dig out the different attack patterns of NSL-KDD dataset (both train and test sets). The abnormal (also called attack) is divided into four categories (Denial of Services (DoS) Attack, Remote to Local (R2L) Attack, User to Root (U2R) Attack, and probe). These attacks are also presented into 2-dimensional space to figure out how they are overlapped and separated from each other. These give the clear pattern of different attacks in host and network-level, and are shown in the Figure 2 (a-f). The goal is to minimize the distance within a class and maximize the distance among the classes. We represented them into three level: normal and attack (two classes) in the Figure 2 (a-b), only attacks (4 classes) in the Figure 2 (c-d), and normal and 4-attack classes in the Figure 2 (e-f). In the next stage, higher fitness features are fed to the classifiers to measure the performance of the system. An overview of the proposed model is shown in the Figure 1.

The organization of the paper is as follows: the description of dataset section I, section II describes the feature selection and discriminant analysis, section III explains the experimental results and discussion. Finally, section IV describes the conclusion of proposed method and future work for intrusion detection and classification.

## II. Data Set

We took the well-known benchmark NSL-KDD dataset [9] for validation of the proposed model. This dataset is a refined version (removed large number of redundant and irreverent samples) of the KDDCup'99 intrusion dataset [11]. It divided into two classes i.e. normal and attack (attack class also divided into 4 sub-classes: DoS attack, R2L attack, U2L attack, and Probe. The detailed description of the data set is shown in Table 1. It contains 41 different features (i.e. shown in Table 2), and features of this dataset type are shown in Table 3. The train set and test set samples are disproportionately distributed in different classes. NSL-KDD dataset contains 125,973 train sets and 22,544 test set samples respectively.

The features belong to 3 categories:

- Basic features [1-9]: the packet capture files of tcpdump are implemented to extract the basic features from the packet headers, TCP segments, and UDP datagram (except payload).
- Content features [10-23]: features are extracted from full payload of TCP/IP packets rooted on domain knowledge in tcpdump files.
- Time-based traffic features [24-41]: features are extracted with a specific temporal window of two seconds.

## III. Feature Selection and Discriminant Analysis

Feature selection is the process of limiting the number of original features dimension by identifying the most discriminative form from the feature pool [16]. It eliminates the set of redundant and irreverent features according to a given algorithm. The genetic algorithm initializes the population with a random set of features, called chromosomes. Each chromosome is evaluated assessing its ability to predict an output based on the accuracy [13]. In the next cycle, the initial population is replaced with a new set of features from different chromosomes that contribute to achieving higher classification

TABLE III: Features of KDDCUP'99 dataset and their corresponding types

| No. | Name of the Feature | Types of the Feature |
|---|---|---|
| 1 | duration | continuous |
| 2 | protocol_type | symbolic |
| 3 | service | symbolic |
| 4 | flag | symbolic |
| 5 | src_bytes | continuous |
| 6 | dst_bytes | continuous |
| 7 | land | symbolic |
| 8 | wrong_fragment | continuous |
| 9 | urgent | continuous |
| 10 | hot | continuous |
| 11 | num_failed_logins | continuous |
| 12 | logged_in | continuous |
| 13 | num_compromised | continuous |
| 14 | root_shell | continuous |
| 15 | su_attempted | continuous |
| 16 | num_root | continuous |
| 17 | num_file_creations | continuous |
| 18 | num_shells | continuous |
| 19 | num_access_files | continuous |
| 20 | num_outbound_cmds | continuous |
| 21 | is_host_login | symbolic |
| 22 | is_guest_login | symbolic |
| 23 | count | continuous |
| 24 | srv_count | continuous |
| 25 | serror_rate | continuous |
| 26 | srv_serror_rate | continuous |
| 27 | rerror_rate | continuous |
| 28 | srv_rerror_rate | continuous |
| 29 | same_srv_rate | continuous |
| 30 | diff_srv_rate | continuous |
| 31 | drv_diff_host_rate | continuous |
| 32 | dst_host_count | continuous |
| 33 | dst_host_srv_count | continuous |
| 34 | dst_host_same_srv_rate | continuous |
| 35 | dst_host_diff_srv_rate | continuous |
| 36 | dst_host_same_src_port_rate | continuous |
| 37 | dst_host_srv_diff_host_rate | continuous |
| 38 | dst_host_serror_rate | continuous |
| 39 | dst_host_srv_serror_rate | continuous |
| 40 | dst_host_rerror_rate | continuous |
| 41 | dst_host_srv_rerror_rate | continuous |

TABLE IV: Selected features by using GA

| No. | Selected Features | No. | Selected Features |
|---|---|---|---|
| 1 | duration | 13 | is_host_login |
| 2 | src_bytes | 14 | count |
| 3 | dst_bytes | 15 | serror_rate |
| 4 | land | 16 | rerror_rate |
| 5 | wrong_fragment | 17 | srv_rerror_rate |
| 6 | urgent | 18 | diff_srv_rate |
| 7 | hot | 19 | drv_diff_host_rate |
| 8 | num_compromised | 20 | dst_host_same_srv_rate |
| 9 | num_file_creations | 21 | dst_host_diff_srv_rate |
| 10 | num_shells | 22 | dst_host_serror_rate |
| 11 | num_access_files | 23 | dst_host_srv_serror_rate |
| 12 | num_outbound_cmds | 24 | dst_host_rerror_rate |

accuracy. This process continues till the desired accuracy is achieved. The list of selected features are shown in Table IV.

The discriminative features (i.e. selected by GA) are further analyzed for class label separability using the Linear Discriminant Analysis (LDA) [8]. The goal is to minimize the distance within a class and maximize the distance among the classes. The higher fitness features are fed to the classifiers to measure the performance of the proposed system. An overview of our model is shown in Figure 1. First, the normal and attack patterns of both training and testing sets are presented into 2-dimensional space to demonstrate the class separability (in Figure 2) (a-b). We only decomposed training and testing set attacks (excluding normal samples) to find out the overlapping patterns of each attack. The abnormal (also called attack) is divided into four categories (DoS attack, R2L attack, U2R attack, and probe). Among them, DoS and Probe are highly separated than R2L and U2R (they are overlapped each other). The class label separability of both train and test set attack patterns are presented in Figure 2 (c-d). Finally, we considered all attacks (DoS, R2L, U2R, and Probe) and normal, and presented in 2-dimensional space. The results of 5-classes are shown in Figure 2 (e-f).

## IV. Experimental Results and Discussions

We calculated the different parameters (accuracies, ROC, etc.) to verify the consistency of the intrusion detection system. We measured the performance of the proposed technique based on selected features (i.e. GA) using SVM. The outcomes of the classifiers recorded the same methodology of discriminant analysis (in three-level).

1) Accuracy: it estimates the ratio of risk recognized of the entire conditions (cases). If accuracy is higher, the machine learning model is better.

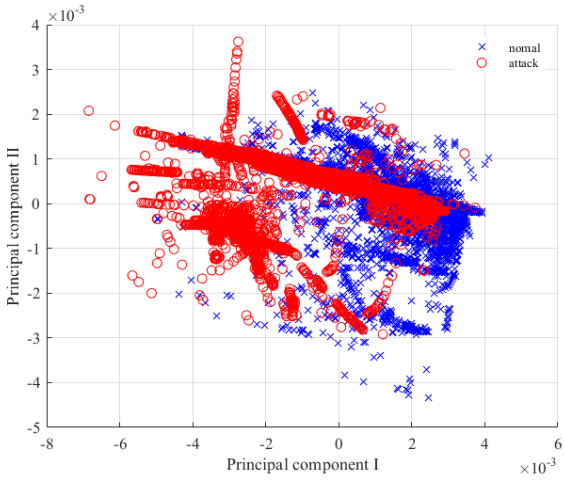
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \qquad (1)$$

The classification accuracies of training and testing sets were 97.0% and 93.0% using SVM (in two classes, normal and attack). Similarly, the overall accuracies of training and testing set (only) attacks categories were 95.75% and 87.50%. Finally, we also measured the classification accuracies between the normal and 4-attack classes (total 5-classes) train and test sets and recorded 95.0% and 91.0%, respectively. We also compared the output of the proposed method with other well-known approaches. Among them, our method outperformed, and the comparison result are shown in Table V.

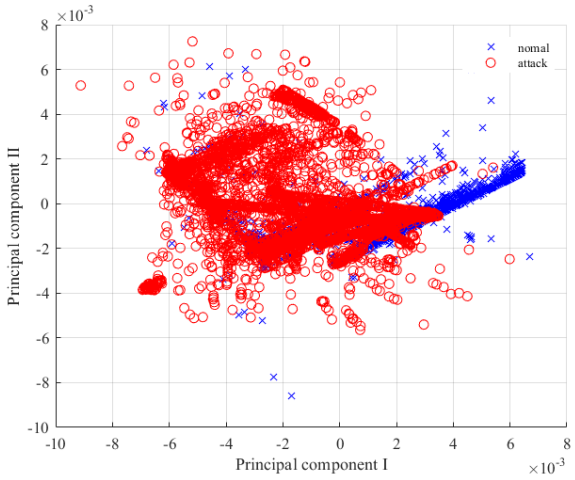
- True Positive ($TP$) - test result is one that detects the risk when the risk is present.
- True Negative ($TN$) - test result is one that does not detect the risk when the risk is absent.
- False Positive ($FP$) - test result is one that detects the risk when the risk is absent.
- False Negative ($FN$) - test result is one that does not detects the risk when the risk is present.

In the next stage, we also plotted the Receiver Operating Characteristic (ROC) curve. It is a graphical representation of true positive rate (TPR) (in y-axis) against its false positive rate (FPR) (in x-axis) [15] [16]. In another way, it is a plot of sensitivity vs (1-specificity) for different cut points, and the area under the curve (AUC) represents the performance of the classifier. The curve close to the upper left corner (TPR) means

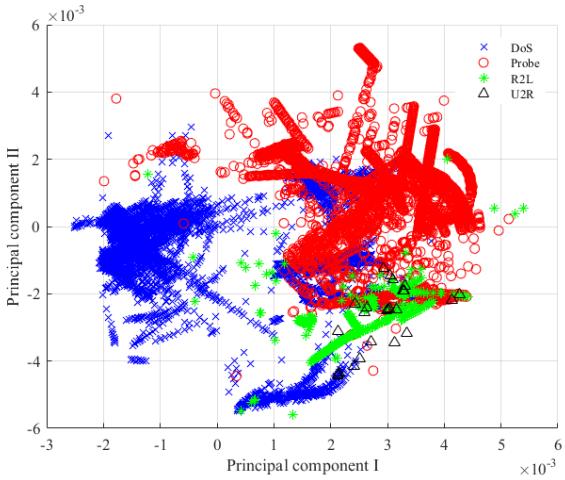
ISBN 979-11-88428-06-9



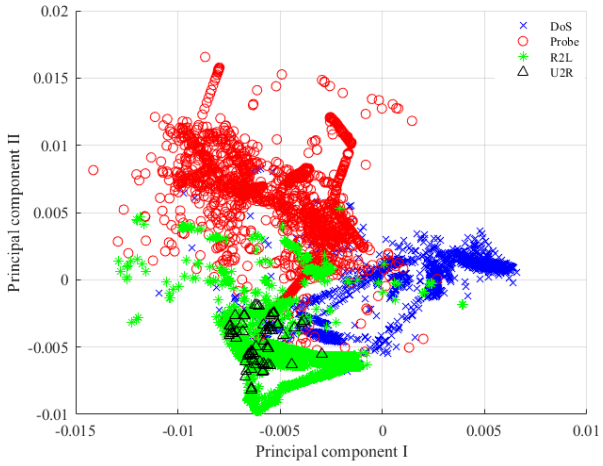

(a) Training set intrusion features representation in 2-dimensional space having normal and attack categories



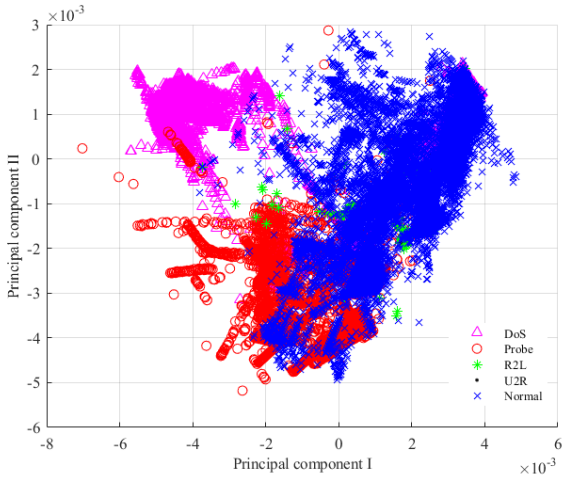

(b) Testing set intrusion features representation in 2-dimensional space having normal and attack categories



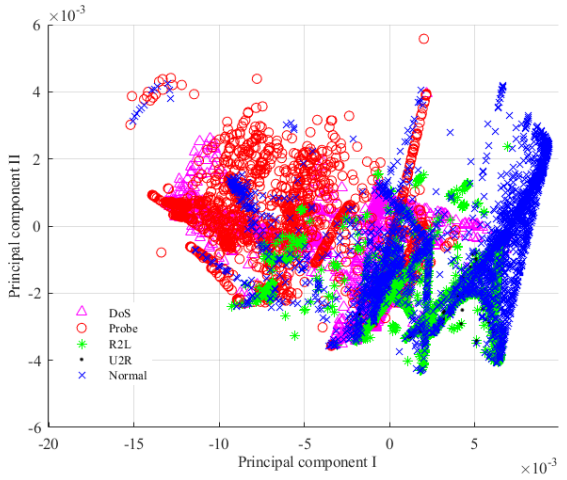

(c) Training set intrusion features representation in 2-dimensional space only attack categories (4-classes)




(d) Testing set intrusion features representation in 2-dimensional space having normal and four different attack categories (4-classes)

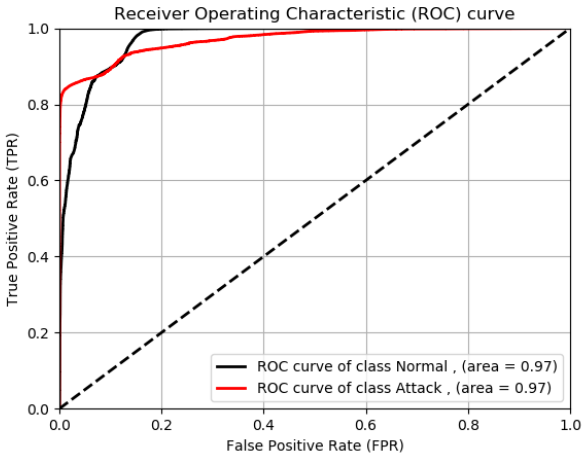



(e) Training set intrusion features representation in 2-dimensional space having normal and four different attack categories (5-classes)

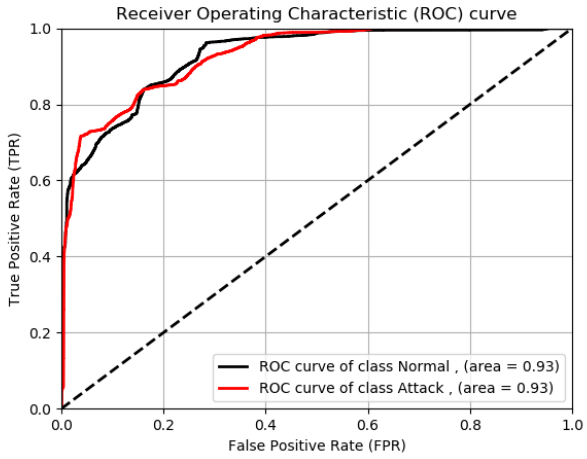



(f) Testing set intrusion features representation in 2-dimensional space having normal and four different attack categories (5-classes)

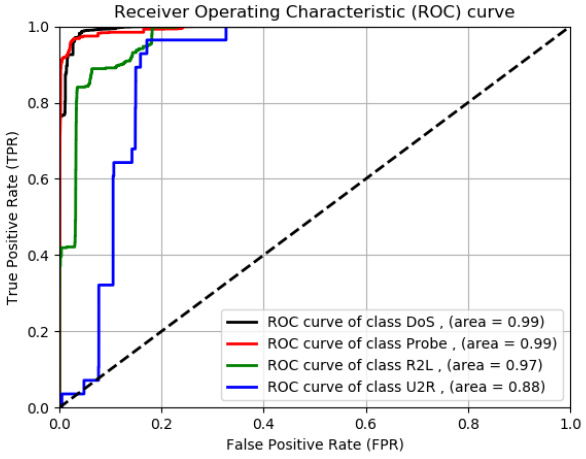
Fig. 2: Discriminant representation of both training and testing set of NSL-KDD dataset using LDA



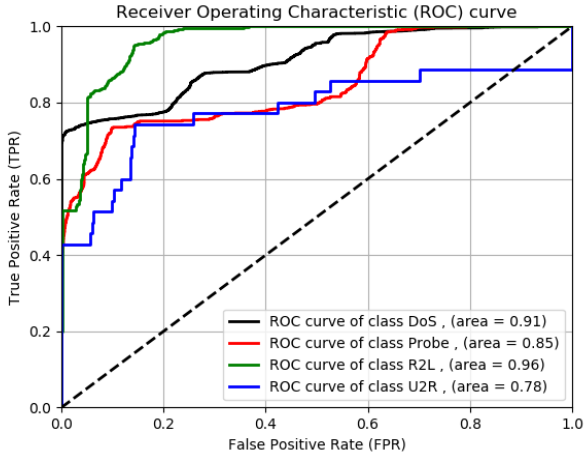

(a) The receiver operating characteristic curve of NSL-KDD training dataset and it has two categories (i.e. normal and attack)



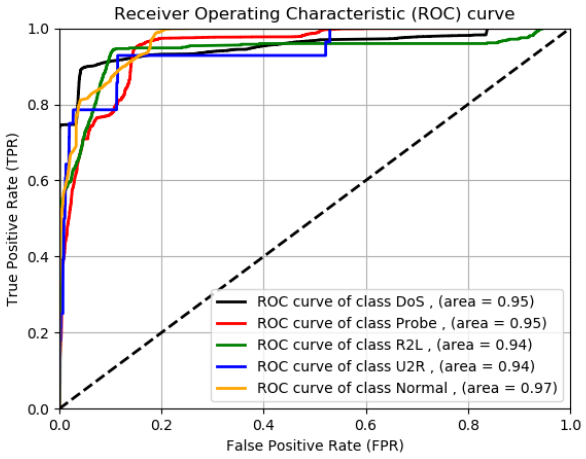

(b) The receiver operating characteristic curve of NSL-KDD testing dataset and it also has two categories (i.e. normal and attack)



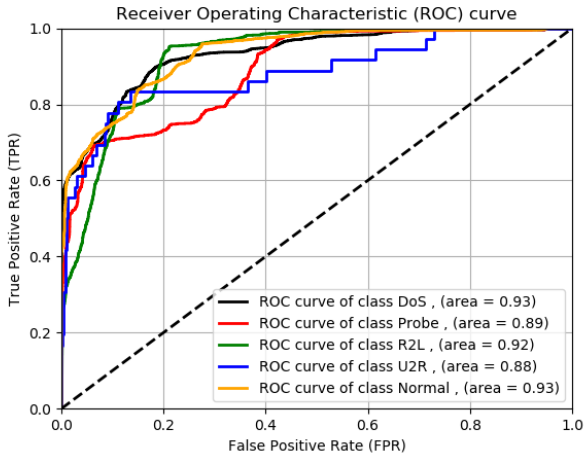

(c) The receiver operating characteristic curve of NSL-KDD training dataset containing only 4 different attacks categories




(d) The receiver operating characteristic curve of NSL-KDD testing dataset containing only 4-different attacks categories




(e) The receiver operating characteristic curve of NSL-KDD of training set containing normal and 4-different attacks categories




(f) The receiver operating characteristic curve of NSL-KDD of testing set containing normal and 4-different attacks categories

Fig. 3: The receiver operating characteristic curve of training and testing sets of NSL-KDD dataset

TABLE V: Comparison of classification accuracies of proposed method with other well-known approaches

| No. | Method and Classifer | Accuracy in % |
|---|---|---|
| 1 | J48 | 81.05 |
| 2 | Naive Bayes | 76.56 |
| 3 | NB Tree | 82.02 |
| 4 | Random Forest | 80.67 |
| 5 | Random Tree | 81.59 |
| 6 | Multi-layer Perceptron | 77.41 |
| 7 | SVM | 69.52 |
| 8 | Proposed method (normal vs attack) | 93.00 |

the diagnostic test has high discriminatory ability. If the curve is close to or below the diagonal, it means the diagnostic test has high discriminatory ability. We plotted the ROC curve in three different stages of discriminant analysis (in Figure 3). The normal vs attack (training and testing sets) ROC curves are shown in Figure 3 (a-b). Similarly, all attacks (train and test sets) ROC curves shown in Figure 3 (c-d) and normal vs all attacks in Figure 3 (e-f).

## V. Conclusion

The proposed IDS is designed to achieve optimal accuracy based on minimal information. It only selected the discriminatory features using GA from the feature pool. The discriminatory features plotted in 2-dimensional space, and their distribution showed how the different (normal and attack) patterns overlapped and separated each other. In the next stage, discriminatory features are fed to the classifier to measure the output of the system. IT was analyzed in three different categories: normal and attack (2-classes), 4-different attacks (excluding normal) classes, and normal and 4-attacks (total 5-classes). We obtained best accuracies in all stages in both training and testing sets. The classification accuracies of training and testing sets were 97.0% and 93.0% using SVM (in two classes). Similarly, the overall accuracies of training and testing sets (only) attacks categories were 95.75% and 87.50%, and the normal and all attacks classes training and testing sets were 95.0% and 91.0%.

In future, we are going to measure the outcome of selected features using different classifiers. Besides these, more feature reduction and selection algorithms will be employed to analyze the impact of discriminatory features for intrusion detection systems.

## References


[1] P. Mishra, V. Varadharajan, U. Tupakula and E. S. Pilli, "A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection," IEEE comm. surveys and tutorials, vol. 21, no. 1, pp. 686-728, 2019.

[2] C. Sinclair, L. Pierce and S. Matzner, "An application of machine learning to network intrusion detection," In Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual (pp. 371-377). IEEE, 1999.

[3] D. P. Benjamin, P. Pal, F. Webber, P. Rubel and M. Atigetchi, "Using a cognitive architecture to automate cyberdefense reasoning," In Bioinspired Learning and Intelligent Systems for Security, 2008. BLISS'08. ECSIS Symposium on (pp. 58-63). IEEE, 2008, August.

[4] L. Zomlot, S. Chandran, D. Caragea and X. Ou, "Aiding intrusion analysis using machine learning," In Machine Learning and Applications (ICMLA), 2013 12th International Conference on (vol. 2, pp. 40-47). IEEE, 2013, December.

[5] M. Rege and R. K. Mbah, "Machine Learning for Cyber Defense and Attack," in 7th International conference of Data Analytics, Athens, Greece, 2018.

[6] S. B. Kotsiantis, I. Zaharakis and P. Pintelas, "Supervised machine learning: A review of classification techniques," Emerging artificial intelligence applications in computer engineering, vol. 160, pp. 3-24, 2007.

[7] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu and C. Wang, "Machine learning and deep learning methods for cybersecurity," IEEE Access, vol. 6, pp. 35365-35381, 2018.

[8] B. K. Baniya and J. Lee, "Importance of audio feature reduction in automatic music genre classification," Multimed Tools Appl 75, 3013–3026 (2016). https://doi.org/10.1007/s11042-014-2418-z

[9] M. Tavallaee, E. Bagheri, W. Lu and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in 2nd IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

[10] B. K. Baniya and E. Z. Gnimpieba, "The Effectiveness of Distinctive Information for Cancer Cell Analysis Through Big Data," Advances in Computer Vision (CVC) 2019, pp. 57-68, vol 944, https://doi.org/10.1007/978-3-030-17798-0_7

[11] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the jam project," discex, vol. 02, pp. 1130, 2000.

[12] B. K. Baniya, J. Lee and Z. Li, "Audio feature reduction and analysis for automatic music genre classification," 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, 2014, pp. 457-462.

[13] A. Garca-Dominguez, C. E. Galván-Tejada, L. A Zanella-Calzada, H. Gamboa-Rosales, J. I. Galván-Tejada,J.M. Celaya-Padilla, H. Luna-Garca and R. Magallanes-Quintanar, "Feature Selection Using Genetic Algorithms for the Generation of a Recognition and Classification of Children Activities Model Using Environmental Sound," Mobile Information Systems, Hindawi, 2020

[14] B. K. Baniya, D. Ghimire and J. Lee, "Automatic music genre classification using timbral texture and rhythmic content features," 2015 17th International Conference on Advanced Communication Technology (ICACT), Seoul, 2015, pp. 434-443, doi: 10.1109/ICACT.2015.7224907.

[15] J. N. Mandrekar, "Receiver Operating Characteristic Curve in Diagnostic Test Assessment," Journal of Thoracic Oncology, vol. 5, no. 9, pp. 1315-1316, 2010

[16] B. K. Baniya, C. Lushbough and E. Z. Gnimpieba, "Significance of reduced features for subcellular bioimage classification," In: International Symposium on Bioinformatics Research and Applications (ISBRA), Minsk, Belarus, 2016


**Babu Kaji Baniya** received the B.E. degree in Computer Engineering from Pokhara University, Nepal in 2005 and M.E. and Ph.D. in Electronic Engineering and Computer Science and Engineering from Chonbuk National University, Republic of Korea in 2015. He also worked as a postdoctoral researcher in the Department of Computer Science and Biomedical Engineering at the University of South Dakota. Currently, he is an assistant professor in the Department of Computer Science and Digital Technologies at Grambling State University, Louisiana, United States of America. His main research interest includes audio signal processing, information retrieval, cybersecurity, bioinformatics, Big Data, machine learning, etc.