IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Double-Layered Hybrid Approach for Network Intrusion Detection System using combined Naive Bayes and SVM

TREEPOP WISANWANICHTHAN[1], MASON THAMMAWICHAI[1]
[1]Navaminda Kasatriyadhiraj Royal Thai Air Force Academy, Bangkok, 10220 Thailand

Corresponding author: Treepop Wisanwanichthan (e-mail: treepop@rtaf.mi.th).

This work was supported in part by the Royal Thai Air Force Cyber Security Research Program under Grant 0606.3/36.

ABSTRACT A pattern matching method (signature-based) is widely used in basic network intrusion detection systems (IDS). A more robust method is to use a machine learning classifier to detect anomalies and unseen attacks. However, a single machine learning classifier is unlikely to be able to accurately detect all types of attacks, especially uncommon attacks e.g., Remote2Local (R2L) and User2Root (U2R) due to a large difference in the patterns of attacks. Thus, a hybrid approach offers more promising performance. In this paper, we proposed a Double-Layered Hybrid Approach (DLHA) designed specifically to address the aforementioned problem. We studied common characteristics of different attack categories by creating Principal Component Analysis (PCA) variables that maximize variance from each attack type, and found that R2L and U2R attacks have similar behaviour to normal users. DLHA deploys Naive Bayes classifier as Layer 1 to detect DOS and Probe, and adopts SVM as Layer 2 to distinguish R2L and U2R from normal instances. We compared our work with other published research articles using the NSL-KDD data set. The experimental results suggest that DLHA outperforms several existing state-of-the-art IDS techniques, and is significantly better than any single machine learning classifier by large margins. DLHA also displays an outstanding performance in detecting rare attacks by obtaining a detection rate of 96.67% and 100% from R2L and U2R respectively.

INDEX TERMS Correlation Feature Selection, Double-Layered Hybrid Approach, Machine Learning, Naive Bayes, Intrusion Detection System, Network Security, NSL-KDD, SVM

I. INTRODUCTION

Due to a dramatic increase of attacks on machines and network-based services, cyber security has become an essential topic in protecting systems from threats at a local and global scale over the past decades. Although network firewalls and data encryption have already provided basic security for computers and networks, as well as satisfied the requirements of fundamental security, there are still a large number of threats that have gone unnoticed and given rise to detrimental effects on the services as a whole [1], [2]. Intrusions are dangerous threats that require immediate attention. Intruders pose the greatest risk to organizations, particularly to units that require a high level of security such as military bases and airports. Failure to detect intruders inevitably leads to security breaches such as the theft of classified information, gaining unauthorized access, and disguising as an administrator for destructive purposes [2]. According to NSL-KDD [3], there are four major classes of attacks 1) Denial Of Service (DOS) is an attack that floods the target with a massive amount of traffics in order to render the service unavailable abruptly. 2) Probe is an attack that scans and exploits network vulnerabilities in open ports to identify services run by the target. 3) Remote2Local (R2L) is an attack that attempts to exploit the target's vulnerabilities to gain illegal access to local networks. 4) User2Root (U2R) is an attack that attempts to exploit the vulnerabilities of the machine to gain root privileges or take over control of the machines. R2L and U2R attacks are uncommon but pose a more detrimental effect to a system [4].

In recent years, Intrusion Detection System (IDS) increasingly plays a vital role in discovering malicious activities due to a massive expansion of network-connected IT devices around the world [5]. IDS methods can be classified as a signature-based (misuse) method and an anomaly-based method. While the signature-based method is able to detect only known malicious activities but not the novel ones, an anomaly-based method offers a better solution that is capable of detecting unknown attacks including potential zero-day exploits. It works by observing a deviation from normal traffic patterns [2]. The signature-based IDS works by matching the traffic target with the pre-defined signatures e.g., Snort [6], in this way, it is very accurate in finding known threats. However, it is utterly worthless in the case of unknown threats [2]. Thus, advanced techniques for the anomaly-based IDS need to be explored [7]. Even though anomaly-based IDS usually produce high false alarm rates [2], nowadays it has gained widespread acceptance amongst the IDS research community [8], [9]. One of the best options in the domain is to use a Machine Learning (ML) approach to create an effective model in order to build a pattern recognition of intruders [1], [8], [9].

Various machine learning techniques have been explored and implemented to build an anomaly-based IDS [10]–[18]. There are two ML techniques that are widely implemented in the IDS field 1) Supervised learning, which creates a mapping function based on pre-defined input-output pairs, and 2) Unsupervised learning, which allows a model to discover internal relationships by itself. Supervised ML is the most widely used technique in IDS. For example, Support Vector Machine (SVM) [10], [16], [19], [20], Decision Tree (DT) [21], [22], K-Nearest Neighbors (KNN) [17], [23], and Naive Bayes (NB) [18], [24]–[26]. Unsupervised ML mostly refers to clustering algorithms such as K-Means [27].

The key challenge in building an efficient IDS is the selection of relevant features in the case of multiple attack categories. Moreover, there will likely be many attack types in networks for ML to learn. Thus, Feature Selection (FS) is a crucial process to eliminate uninformative attributes and noise. FS is one of the primary factors to enhance accuracy in IDS [13], [28]. Thus, many IDS researchers try to explore the best feature selection methods to extract a subset of relevant features in order to boost classification results [29] such as using Local Search Algorithm with K-Means [13], Genetic Algorithm (GA) [19], [28], Particle Swarm Optimization (PSO) [28], Ant Colony Algorithm [28], [30], and Correlation Coefficient [31]–[33]. In the past years, Artificial Neural Networks (ANN) and Deep Learning (DL) have been successfully applied to deal with complex patterns, especially in image and language processing. There are studies that utilized ANN on IDS problem such as Convolutional Neural Networks (CNN) [34]–[36], Recurrent Neural Networks (RNN) [37].

Every ML algorithm has its own capability. One can precisely detect a specific type of attack, while others are not accurate at it [38], [39]. Techniques that combine two or more learning algorithms have been recently proposed due to superior performance in detecting various attacks [39]. Ensemble method is a popular learning algorithm for IDS, which usually offers a better result over a single estimator [11]. Ensemble learning technique is the process where multiple base classifiers are combined to achieve better predictive capability, for example, Random Forest (RF) [14], [40].

In the past years, another approach that has been adopted largely in the IDS research community is a hybrid approach. A hybrid approach, in general, refers to a method that combines two or more learning techniques e.g., using a signature-based method with an anomaly-based method [41]–[43], or an anomaly-based method with an anomaly-based method. For example, unsupervised ML and supervised ML [38], and supervised ML and supervised ML [28], [44]–[46]. The main concept behind the hybrid approach is to exploit the advantages of each learning technique by combining the strong points of different single classifiers in order to improve the overall detection rate. It is also an effective technique that is used to reduce bias towards more frequent attacks as a result of data set imbalance [46]. Therefore, the hybrid approach is a promising technique to address the major concerns in IDS research.

However, there are three key problems in previous studies. (I) Many works e.g., [37], [47] only focused on using a single machine learning model to detect all attack types. This led to a drawback of a single classifier that is difficult to outperform a hybrid approach. (II) Low-frequency attacks are not well detected due to a severe imbalance of classes in the training data set, which results in bias in ML models [48]. (III) Relevant features for a specific type of attack may not be necessary for other attacks due to a vast difference in attack behaviours [49], [50].

In order to address the above problems, our contributions to the cyber security domain are as follows: (I) We proposed a Double-Layered Hybrid Approach (DLHA) that is better than a single ML classifier and the ensemble method. The proposed approach is composed of two layers that work in a cascading manner, where the first layer is to detect DOS and Probe, and the second layer is to detect R2L and U2R. (II) We performed data analysis using PCA and found that DOS and Probe are more distinct from the rest, and R2L and U2R behave similarly to normal traffic patterns. The findings inspired us to design DLHA. Contributions (I) and (II) are exclusively dedicated to demonstrating the effectiveness of implementing a hybrid approach, as opposed to using one classifier as mentioned in problem (I). (III) The uniqueness of our approach is that we divided the NSL-KDD training data set into two groups i.e., 1) Group 1 that contains all classes, and 2) Group 2 that contains only R2L, U2R, and Normal classes. These were used to separately train the two classifiers in order to have a dedicated classifier for detecting rare attacks i.e., R2L and U2R amongst normal connections. The group-divided strategy allows the algorithm to focus on low-frequency attacks at the second layer to address the problem (II). (IV) We presented Intersectional Correlated Feature Selection (ICFS) using correlation coefficients. It selected commonly important features from different attack types within the subgroups in order to mitigate the problem (III). (V) We conducted an evaluation of our proposed approach to show that DLHA yields higher detection rates on both

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3118573, IEEE Access

overall performance and low-frequency-attack performances compared to many other existing state-of-the-art methods. (VI) We showed that DLHA is highly competitive as a hybrid method, and it has a substantially superior performance to the traditional single ML techniques.

The rest of the paper is organized as follows. In section II, related works on anomaly-based IDS are provided. Data analysis on NSL-KDD is explained and shown in Section III. The conceptual framework of our proposed DLHA is illustrated, and the combined NB and SVM detection system is introduced in Section IV. Section V explains the performance analysis of DLHA as well as presents an extensive comparison of our results to other anomaly-based IDS techniques. A conclusion is provided in Section VI.

II. RELATED WORK

Numerous anomaly-based IDS nowadays implement a hybrid ML model, as it leads to better performance and enhanced efficiency [1], [39]. Chi-square feature selection with multi-class SVM model was proposed in [51]. Chi-square was used to calculate statistical significance on each feature, and then the low-rank features were removed. The number of features decreased from 41 to 31 during the feature selection process. Then, hyperparameter tuning was performed for RBF-kernel SVM to obtain the best combination of parameters i.e., C and gamma. The model led to an outstanding result, but the authors did not perform an evaluation in *KDDTest+*. Yao *et al.* [39] proposed a Hybrid Multi-Level data mining framework using hybrid feature selection. The authors performed several experiments to choose the best ML algorithms to detect each class of attack. The final detection system consisted of four different classifiers, which were: 1. Linear SVM to detect DOS, 2. ANN with logistic activation function to detect Probe, 3. ANN with relu activation function to detect R2L, and 4. ANN with identity activation function to detect U2R. The hybrid framework resulted in a superb performance, but the framework could be cumbersome for a real-time IDS as it consisted of four classifiers. The data fusion method performed better than using a single classifier alone by integrating multiple different classifiers and predicting at the last step. It allows flexibility of data pre-processing by using different feature selection methods. However, the use of different classifiers for different data sources resulted in longer computational time both in training and testing processes [52].

GA-SVM that implemented Genetic Algorithm (GA) combined with SVM was introduced in [53]. The genetic algorithm was used as a feature reduction technique to reduce features from 45 to 10 based on three priorities. The GA applied crossover and variation to generate the optimal subsets of features used in training by SVM. The efficient anomaly-based IDS hybrid model was proposed in [54]. The authors used a voting algorithm with information gain to filter out irrelevant features. The designed hybrid classifier algorithm utilized ensemble representing J48, Meta Tagging, RandomTree, REPTree, AdaBoostM1, DecisionStump, and Naive Bayes. This method claimed to address the high false negative rate. Jiang *et al.* [34] proposed a combined hybrid sampling with a Deep Hierarchical Network model. The model was tasked to balance the class distribution by initially employing One-Side Selection (OSS) to reduce samples in the majority classes, then use Synthetic Minority Over-sampling Technique (SMOTE) to increase the samples in the minority classes. The deep hierarchical network model worked based on spatial feature extraction with Convolution Neural Network (CNN) and temporal feature extraction with Bi-directional Long Short-Term Memory (BiLSTM). The model accurately detected the under-represented classes as a result of a hybrid sampling technique.

Biswas *et al.* [55] proposed hybrid feature selection with neural network and K-Means clustering. It applied PCA to K-Means clustering, which specified five clusters as per the number of classes. Each cluster was trained and evaluated by aggregating the results from different ANN functions i.e., feed forward neural network algorithm. Mazini *et al.* [56] proposed a new hybrid anomaly-based IDS framework to improve detection rates using Artificial Bee Colony (ABC) as a feature selection technique and AdaBoost algorithm as a classifier. The authors implemented an ABC meta-algorithm to select the best subset of relevant features and deployed AdaBoost.M2 to detect multi-class attacks. The IDS based on Naive Bayes Classifier (NBC) using Bayesian probability was presented in [57]. The NBC calculated probabilities of any attack occurrence and the TCP normal traffic based on the Bayesian network. The authors performed a score map analysis to select the features that boost detection rates. The results of NBC improved the detection rate of R2L attacks. Çavuşoğlu [58] introduced a new hybrid IDS, which used a combination of different classifiers and feature selection techniques according to each type of attack. The authors performed CfsSubsetEval and WrapperSubsetEval feature selection according to protocol types on the different feature selection algorithms. The proposed IDS works in a multi-level manner by having four different techniques for each attack class i.e., RF to detect DOS, Stacking method with RF, J48, and KNN to detect R2L, RF to detect Probe, and J48 and NB to classify normal traffics and U2R.

Hwang *et al.* [59] presented the three-tier architecture IDS approach by implementing a blacklist, whitelist, and SVM. The first tier was to filter out the known attacks, the second tier was to classify normal connections, and the last tier was to detect anomalies from the rest of the connections. The authors claimed that the method was efficient and flexible as all connections were not passed to every tier process. Pajouh *et al.* [60] proposed a Two-layer Dimension reduction and Two-tier Classification model (TDTC) to focus on detecting malicious activities i.e., R2L and U2R. The authors' framework utilized two dimensionality reduction techniques: PCA and Linear Discriminate Analysis (LDA). After PCA, LDA was applied with labels to transform data into lower dimensions in order to have as few dimensions as possible to suit the IoT environment. At the two-tier classification system, NB and Certainty Factor of the KNN algorithm were deployed. Tama *et al.* [28] presented a Two-Stage Ensemble (TSE-IDS) model that performed three feature selection algorithms i.e., Particle Swarm Optimization (PSO), Ant Colony Algorithm (ACO), and Genetic Algorithm (GA). The features were selected based on the performance of the pruning tree classifier (REPT). The two-stage meta classifier was proposed using rotating forest and bagging to perform the majority voting at the end. The predictive features, as a result of the three feature selection algorithms, were used in training. Then, a 10-fold CV was used to measure average accuracy in the training set at the validation stage. The results suggested that a hybrid approach performed relatively better than single ML classifiers.

Alfantookh [61] introduced Denial of Service Intelligent Detection (DoSID), which used feed forward ANN with the backpropagation algorithm to detect DOS attacks. The author presented the Grey Area that used the distribution concept and conducted experiments to evaluate different parameter sets to select the best configurations for ANN, such as the number of training epochs. The experimental results displayed a capability to detect unknown attacks that have never been seen at the training process, as well as an improvement in false negative rates. A two-tier classifier with LDA feature selection was introduced in [4]. The model was trained on the training data set that applied SMOTE to make the data set more balanced in terms of the ratio between anomalies and attack records. The NB and KNN classification algorithms were employed in the proposed IDS system. Compared to other papers, it achieved a high detection rate on uncommon attacks such as R2L and U2R.

Baykara and Das [62] proposed a hybrid honeypot based real-time intrusion detection and prevention system. The system was developed by utilizing low and high interaction honeypots to reduce installation, configuration, maintenance and management cost. The approach led to a considerable drop of a false positive rate, which benefited the real-time enterprise network monitoring. An adaptive ensemble ML IDS framework was presented in [11]. The authors proposed a MultiTree algorithm to deal with skewed class distribution in the training set. It adjusted a proportion of the training data set in order to reduce bias towards over-represented classes. The authors evaluated multiple classifiers to select the base classifiers including Decision Tree, Random Forest, KNN, and Deep Neural Networks. In the end, adaptive majority voting was used to make a final prediction. However, the results indicated a high false alarm rate, especially on Probe attacks. A hybrid approach using a two-step binary classification method was demonstrated in [46]. The authors designed the first step to be an ensemble algorithm by deploying several binary classifiers with one aggregation function to predict the exact class of the connection. The second step was based on the outcome of the first step by performing the KNN algorithm to predict its class when the first step failed to confirm a certain class. This hybrid approach accomplished a satisfactory performance in detecting rare attacks i.e., R2L and U2R.

De la Hoz *et al.* [63] proposed a hybrid framework using PCA, Fisher Discriminant Ratio (FDR), and Probabilistic Self-Organizing Maps (PSOMs). PCA was used to extract meaningful components from all data attributes, and FDR was considered as a feature selection to maintain informative features. The PSOMs algorithm was used to detect anomalous instances. A fuzzy anomaly-based IDS with Content-Centric Networks was introduced in [64]. The approach hybridized the PSO and K-Means algorithm to optimize the proper number of clusters obtained from performing K-Means. At the classification stage, the fuzzy algorithm was deployed to distinguish abnormal connections from normal connections. Auto-Encoder (AE) intelligent IDS was proposed in [65]. The authors performed feature selection by removing features that contain zeros higher than 80%. The rest features combined with resulted features from one-hot encoding were used as feature vectors. The AE was trained in an unsupervised manner using the Scaled Conjugate Gradient method (SCG) for 100 epochs. The authors tested the model with several shallow ANN such as Multi-Layer Perceptron (MLP) and deep ANN such as LSTM.

Recurrent Neural Network (RNN) based IDS was introduced in [37]. The authors implemented one-hot encoding and optimized parameters by adjusting hidden nodes and the learning rate. The model performed well on frequent attacks but not on uncommon attacks because no extra work was done to address the data set imbalance. Honeypot-based intrusion detection and prevention system combined with a software-defined switching was presented in [66]. The system was evaluated in a simulation environment, where the results indicated a reduced false alarm rate. The honeypot server that worked alongside the intrusion detection system, produced signatures of potential zero-day attacks that benefited anomaly-based IDS to detect future unseen attacks more precisely. Gogoi *et al.* [38] proposed a Multi-Level Hybrid (MLH-IDS) data mining technique. It has three levels where it utilized a supervised ML CatSub+ as the first level to classify DOS and Probe, an unsupervised ML K-point algorithm as the second level to detect normal traffics, and an outlier-based classifier GBBK as the third level to classify R2L and U2R. MLH-IDS produced excellent results as a hybrid technique in detecting all types of attacks using NSL-KDD. However, its real performance remains unclear because the authors marked the attacks that exist in *KDDTest+*, but not in *KDDTrain+*, as unknown in the testing process.

Bostani and Sheikhan [67] proposed a graph-based ML framework based on a modified Optimum-path Forest model (OPF). In the framework, the authors used K-Means to partition the original NSL-KDD data set into K different training subsets, which are used in the training process of OPFs. The concept of centrality and prestige in social network analysis was employed in a pruning module to extract the most predictive samples from the subsets obtained by implementing K-Means to accelerate the OPF stage. Instead of using the full features, Mohammadi *et al.* [33] proposed a group-based feature selection, which was called Feature Grouping based on Linear Correlation Coefficient (FGLCC) combined with CutterFish Algorithm (CFA) on clustering of different groups. FGLCC measured linear correlation coefficients from features and classes to select the maximum correlation in order to reduce computational cost in a large sample size. The algorithm improved the accuracy and the detection rate of IDS. Pervez and Farid [47] developed an anomaly-based IDS using SVM with the proposed feature selection algorithm. The feature selection algorithm kept removing one input feature, then built a classifier to test if a new subset of features led to better classification accuracy. The best classification accuracy was obtained by using 41 features, where it achieved 98.96% from a 10-fold CV in *KDDTrain+*. However, it experienced a major drop in the accuracy down to 82.37% when tested with *KDDTest+*.

Considering past related works, the key difference amongst hybrid approaches is feature selection. While many methods perform feature selection based on the most relevant features to all attacks, the better alternative is to perform feature selection on a specific attack type. For example, a hybrid feature selection for each hybrid level was used in [39]. Another major difference is a hybrid design. In [39], [58], the authors employed four classifiers to detect each type of attack, which led to better performance but a slower process. On the other hand, Pervez and Farid [47] presented a two-tier hybrid IDS using two classifiers with optimal features derived from PCA and LDA. However, the two-tier IDS met an inefficiency in the R2L detection performance. Thus, past papers have failed to make contributions in effective feature selection, and more efficient hybrid IDS design. Table 1 highlights key differences and a summary of the closest related works to our study that proposed a hybrid approach. The summary explains feature selection, ML algorithm, evaluation criteria, and the main contribution, including our work.

III. DATA ANALYSIS

A. DATA SET DESCRIPTION

KDD99 [70] was the most widely used data set in evaluating anomaly-based IDS approaches [71], it captured TCP dump data from DARPA98 off-line intrusion detection evaluation program. However, the KDD99 has numerous inherent problems. Hence, NSL-KDD data set [3] is instead utilized in this paper. The NSL-KDD was proposed in 2009 to solve the KDD99 data set that is skewed, and disproportionately distributed [3]. The advantages and improvements that the NSL-KDD holds over the outdated KDD99 are that a huge number of redundant/duplicated data are removed. Also, selected instances are well represented i.e., the numbers of attacks and normal instances are not very distinct, and the difficulty levels of attacks are evenly distributed in the training and testing sets. This results in more reliable classification results when comparing anomaly-based methods using different ML techniques [1], [3], [72].

In addition, it also alleviates bias in the evaluation stage, which originally caused a higher detection rate towards frequent attacks [3]. Therefore, NSL-KDD is the standardized data set used by a number of network IDS researchers [1], [28], [34], [65], [72]–[74]. In this paper, we only consider three data sets, which are *KDDTrain+*, *KDDTrain+_20Percent*, and *KDDTest+*. *KDDTrain+_20Percent* is a subset of *KDDTrain+*, which contains 20% of instances with the same distribution ratio of classes. The reason behind the selection of the three data sets is that we can perform an extensive evaluation of our algorithm using *KDDTest+* that contains 17 unseen attack classes. The training is done by utilizing the full sample size in *KDDTrain+* data set first, then a comparatively smaller size i.e., *KDDTrain+_20Percent* data set in order to observe the difference in performance when the training data are relatively smaller. According to NSL-KDD, there are four main categories of attacks as shown in Table 2.

The NSL-KDD consists of five classes i.e., DOS, Probe, R2L, U2R, and Normal. The detailed distribution of five classes in *KDDTrain+*, *KDDTrain+_20Percent*, and *KDDTest+* are displayed in Table 3 and Table 4 respectively. Although the NSL-KDD is an updated version of KDD99, it still suffers from an inherited uneven class distribution within the data sets. For example, in the training data set it is observed that normal records take the highest share amongst all instances, which is about 53.46% in training data followed by DOS (36.46%), and Probe (9.25%) while R2L (0.79%) and U2R (0.04%) sample data are very scarce. The problem is that if a single model is deployed, it will not be able to detect R2L and U2R effectively owing to the model's bias [72]. R2L and U2R attacks, used by hackers, are more harmful than DOS and Probe [4].

Furthermore, it is also evident that the discrepancy of the numbers of R2L between training and testing is very high i.e., R2L takes up to 22.48% of all attacks in testing data, but only 1.70% in training data. Hence, in order to enhance overall IDS performance, R2L attacks need to be well detected. It is worth noting that the testing data set (*KDDTest+*) contains 17 additional unseen minor classes of attacks, which do not appear in the training data set before i.e., *apache2, httptunnel, mailbomb, mscan, named, processtable, ps, saint, sendmail, snmpgetattack, snmpguess, sqlattack, udpstorm, worm, xlock, xsnoop,* and *xterm*. Making it more challenging and realistic to assess our hybrid approach against both known and unknown categories of attacks. However, there are two minor classes of attacks that appear in the training data, but they are absent in the testing data set i.e., *spy* and *warezclient.*

B. CLASS DISTRIBUTION ANALYSIS

Each instance in the NSL-KDD contains 41 features as displayed in Table 5. The features can be divided into four categories which are: 1. Intrinsic features (feature 1 to 9) derived from the header of the packets, 2. Content features (feature 10 to 22) contain original packet payloads, 3. Time-based features (feature 23 to 31) extracted from 2-second interval traffic connection records, and 4. Host-based fea-

TABLE 1. Summary of the closest related works that proposed a hybrid approach

Author(s)	Year	Proposed Method	Feature Selection	Algorithm	Evaluation Criteria	Main Contribution
Gogoi *et al.* [38]	2014	MLH-IDS	Mutual Information	CatSub+, *k*-point, and GBBK	Detection Rate FDR	Proposed a 3-level hybrid IDS using supervised, unsupervised, and outlier detection.
Pajouh *et al.* [4]	2016	TDTC	PCA and LDA	NBC and CF-KNN	Detection Rate FAR	Proposed two-layer dimension reduction and two-tier classification
Li *el al.* [46]	2017	Effective Two-Step IDS	Gain Ratio in C4.5	C4.5 DT, KNN	Accuracy F1 Score Precision Detection Rate FAR	Proposed an effective two-step hybrid using binary classification and KNN
Yao *el al.* [39]	2019	HMLD	FMIFS	SVM, ANN	Accuracy F1 Score Precision Detection Rate	Proposed HMLD using hybrid feature selection and hybrid classification
Çavuşoğlu [58]	2019	Hybrid-layered IDS	CfsSubsetEval and Wrapper-SubsetEval	RF, J48 DT, KNN	Accuracy F1 Score Detection Rate TPR FAR MCC	Proposed a hybrid-layered IDS using four different classifiers to detect each attack type
Gao *et al.* [11]	2019	Adaptive Ensemble	CART	DT, RF, KNN, DNN, and MultiTree	Accuracy F1 Score Precision Detection Rate	Proposed an adaptive ensemble using multiple ML algorithms, and adaptive voting algorithm
Tama *et al.* [28]	2019	TSE-IDS	PSO+ACO+GA	Rotating Forest and Bagging	Accuracy Precision Detection Rate FAR	Proposed a two-stage meta classifier with majority voting and hybrid feature selection
Golrang *et al.* [68]	2020	Hybrid Multi-Objective Approach	NSGAII-ANN	Random Forest	Accuracy F1 Score Precision Detection Rate FAR	Proposed a hybrid multi-objective approach to address the redundant feature selection issue
Liu *et al.* [69]	2021	Hybrid K-Means+RF	Attribute Ratio	K-Means, RF,~ and CNN+LSTM	Accuracy TPR	Proposed a hybrid IDS using scalable K-Means Random Forest, and CNN+LSTM for anomaly classification
Wisanwanichthan and Thammawichai (This paper)	2021	DLHA	ICFS and PCA	NBC and SVM	Accuracy F1 Score Precision Detection Rate FAR	Proposed a double-layered hybrid IDS using NBC and SVM with ICFS and PCA for feature selection

TABLE 2. Major Categories of attacks in the NSL-KDD data set

Attack Categories	**Description**
Denial Of Service (DOS)	To make the service unavailable by flooding connections
Probe	To gain important data (port scanning)
Remote to Local (R2L)	To gain Super User (root) privileges
User to Root (U2R)	To gain local access from remote machine

tures (feature 32 to 41) are similar to time-based features but include all series of connections instead of a 2-second interval. These features are beneficial to assess attacks that operate longer than the two-second time span. 39 of the features are numerical, and 3 features are categorical, namely *protocol_type*, *service*, and *flag*.

To perform data analysis on training data, we first implemented data pre-processing by assigning numerical label tags from [Normal, DOS, Probe, R2L, U2R] to [0, 1, 2, 3, 4] respectively. Then, we perform one-hot encoding on those categorical features. One-hot encoding is a powerful tool used to maintain predictive information from converting a categorical feature to numerical features. However, it assumes zero relationships from each value in the category [76]. The categorical features have no internal order or relationship. Let *n* be the number of unique values in a feature, one-hot encoder creates *n* new features corresponding to each unique original value, which contain a vector binary representation. 1 is represented as a presence of a value, and the rest are 0 e.g., ICMP protocol is encoded as [1,0,0], and TCP is encoded as [0,1,0]. After data pre-processing, we ended up having 122 features, but *num_outbound_cmds* feature contains only 0, which indicates no predictive power. The

TABLE 3. 5-Class Distribution in *KDDTrain+* and *KDDTrain+_20Percent*

Major Class	Minor Class	Instance (20%)	Total (20%)
1. DOS	back	956 (196)	45,927 (9,234)
	land	18 (1)	
	neptune	41,214 (8,282)	
	pod	201 (38)	
	smurf	2,646 (529)	
	teardrop	892 (188)	
2. Probe	ipweep	3,599 (710)	11,656 (2,289)
	nmap	1,493 (301)	
	portsweep	2,931 (587)	
	satan	3,633 (691)	
3. R2L	ftp_write	8 (1)	995 (209)
	guess_passwd	53 (10)	
	imap	11 (5)	
	multihop	7 (2)	
	phf	4 (2)	
	spy	2 (1)	
	warezclient	890 (181)	
	warezmaster	20 (7)	
4. U2R	buffer_overflow	30 (6)	52 (11)
	loadmodule	9 (1)	
	perl	3 (0)	
	rootkit	10 (4)	
Total Attacks	58,630 (11,743)		
5. Normal	67,343 (13,449)		
Total Instances	125,973 (25,192)		

TABLE 4. 5-Class Distribution in KDDTest+ (* are attack categories that do not appear in the training data)

Major Class	Minor Class	Instance	Total
1. DOS	apache2*	737	7,460 (1,719*)
	back	359	
	land	7	
	mailbomb*	293	
	neptune	4,657	
	pod	41	
	processtable*	685	
	smurf	665	
	teardrop	12	
	udpstorm*	2	
	worm*	2	
2. Probe	ipsweep	141	2,421 (1,315*)
	mscan*	996	
	nmap	73	
	portsweep	157	
	saint*	319	
	satan	735	
3. R2L	ftp_write	3	2,885 (686*)
	guess_passwd	1,231	
	httptunnel*	133	
	imap	1	
	multihop	18	
	named*	17	
	phf	2	
	sendmail*	14	
	snmpgetattack*	178	
	snmpguess*	331	
	warezmaster	944	
	xlock*	9	
	xsnoop*	4	
4. U2R	buffer_overflow	20	67 (30*)
	loadmodule	2	
	perl	2	
	ps*	15	
	rootkit	13	
	sqlattack*	2	
	xterm*	13	
Total Attacks	12,833 (3,750*)		
5. Normal	9,711		
Total Instances	22,544		

feature, as a result, was dropped. Thus, we only considered 121 features in this work. After standardization was carried out, it removed each value by its mean and divided by its standard deviation as shown in (1).

$$Z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}, i = 1, 2, \ldots, n; j = 1, 2, \ldots, d \quad (1)$$

where $\mu_j = \frac{\sum_{i=1}^{n}(x_{ij})}{n}$, $\sigma_j = \sqrt{\frac{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2}{n}}$, n is the number of samples, and d is the number of dimensions.

In order to gain data insight, we attempted to find characteristics between different attack categories by creating visualization to gain an intuition of the class distribution in two dimensions. We selected PCA as a dimensionality reduction to transform large features into a smaller set of uncorrelated linear features. The output still contains most of the variance from its original data [77]. In this way, we can draw a rough idea of how different classes deviate from each other. In PCA, we constructed linear transformation. Let X be a d dimensional vector from the training set. The new number of features is d' where $d' < d$ in order to obtain the first d' principal components, the covariance matrix computation was performed. The covariance matrix is a square matrix given by $C_{i,j} = \sigma(x_i, x_j)$, where $C \in R^{d \times d}$, and d refers to the number of dimensions or features from the initial data matrix X that $X \in R^{n \times d}$. The covariance matrix can be defined as:

$$C = \begin{pmatrix} \sigma(d_1, d_1) & \cdots & \sigma(d_1, d_n) \\ \vdots & \ddots & \vdots \\ \sigma(d_n, d_1) & \cdots & \sigma(d_n, d_n) \end{pmatrix}$$

hence, it can be computed by:

$$C = \frac{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)\left(X_i - \bar{X}\right)^{\mathsf{T}}}{n-1} \quad (2)$$

TABLE 5. Detailed features in NSL-KDD [75]

Group	No.	Feature Name	Description	Type
Intrinsic Features	1	Duration	Length of time of the connection	Continuous
	2	Protocol Type	Type of protocol used in the connection	Categorical
	3	Service	Destination network service	Categorical
	4	Flag	Status of the connection	Categorical
	5	Src Bytes	Number of data bytes transferred from source to destination	Continuous
	6	Dst Bytes	Number of data bytes transferred from destination to source	Continuous
	7	Land	1 If source and destination IP addresses are equal	Discrete
	8	Wrong Fragment	Total number of wrong fragments in this connection	Discrete
	9	Urgent	Number of urgent packets	Discrete
Content Features	10	Hot	Number of hot indicators in the content	Continuous
	11	Num Failed Logins	Count of failed login attempts	Continuous
	12	Logged In	Login Status, 1 if successful	Discrete
	13	Num Compromised	Number of "compromised” conditions	Continuous
	14	Root Shell	1 if root shell is obtained; 0 otherwise	Discrete
	15	Su Attempted	1 if "su root” command attempted or used	Discrete
	16	Num Root	Number of "root” accesses	Continuous
	17	Num File Creations	Number of file creation operations in the connection	Continuous
	18	Num Shells	Number of shell prompts	Continuous
	19	Num Access Files	Number of operations on access control files	Continuous
	20	Num Outbound Cmds	Number of outbound commands in an ftp session	Continuous
	21	Is Hot Logins	1 if the login belongs to the "hot” list	Discrete
	22	Is Guest Login	1 if the login is a "guest” login	Discrete
Time-based Features	23	Count	Number of connections to the same dst host in the past two seconds	Continuous
	24	Srv Count	Number of connections to the same service in the past two seconds	Continuous
	25	Serror Rate	% connections have activated the flag (4) s0, s1, s2 or s3 from (23)	Continuous
	26	Srv Serror Rate	% connections have activated the flag (4) s0, s1, s2 or s3 from (24)	Continuous
	27	Rerror Rate	% connections have activated the flag (4) REJ from (23)	Continuous
	28	Srv Rerror Rate	% connections have activated the flag (4) REJ from (24)	Continuous
	29	Same Srv Rate	% connections to the same service from (23)	Continuous
	30	Diff Srv Rate	% connections to different services from (23)	Continuous
	31	Srv Diff Host Rate	% connections to different dst machines from (24)	Continuous
Host-based Features	32	Dst Host Count	Count for destination host	Continuous
	33	Dst Host Srv Count	Srv-count for destination host	Continuous
	34	Dst Host Same Srv Rate	Same-srv-rate for destination host	Continuous
	35	Dst Host Diff Srv Rate	Diff-srv-rate for destination host	Continuous
	36	Dst Host Same Src Port Rate	Same-src-port-rate for destination host	Continuous
	37	Dst Host Srv Diff Host Rate	Diff-host-rate for destination host	Continuous
	38	Dst Host Serror Rate	Serror-rate for destination host	Continuous
	39	Dst Host Srv Serror Rate	Srv-serror-rate for destination host	Continuous
	40	Dst Host Rerror Rate	Rerror-rate for destination host	Continuous
	41	Dst Host Srv Rerror Rate	Srv-rerror-rate for destination host	Continuous

Following this, we calculated eigenvalues and eigenvectors, $Av = \lambda v$, corresponding to the computed covariance matrix. It then ranked the eigenvectors with the highest eigenvalues to be the first principal component and so on. Thus, d' is the number of dimensions, sorted in descending order, obtained from implementing PCA. For the purpose of illustration, we chose two as the number of the principal components in order to be able to plot their instances separated by classes on a two-dimensional graph. We performed a scatter plot of the two-dimensional PCA analysis on training data as visualized in Fig 1.

In Fig 1, we labelled DOS as orange, Probe as green, R2L as yellow, U2R as red, and Normal as blue. In the top graph, we excluded Normal. Obviously, most DOS and Probe instances are located far from normal instances, while most R2L and U2R attacks overlap with each other and with the normal connections. It means that R2L and U2R intruders shared some characteristics, or in other words, they behave more similarly to each other than those far-away attacks i.e., DOS and Probe. Given the bottom graph, the majority of DOS and Probe attacks are relatively independent to the rest, with a minor overlapping region at the top. Moreover, only few DOS and Probe records overlap normal connections. It is clear why many IDS methods failed to provide accurate

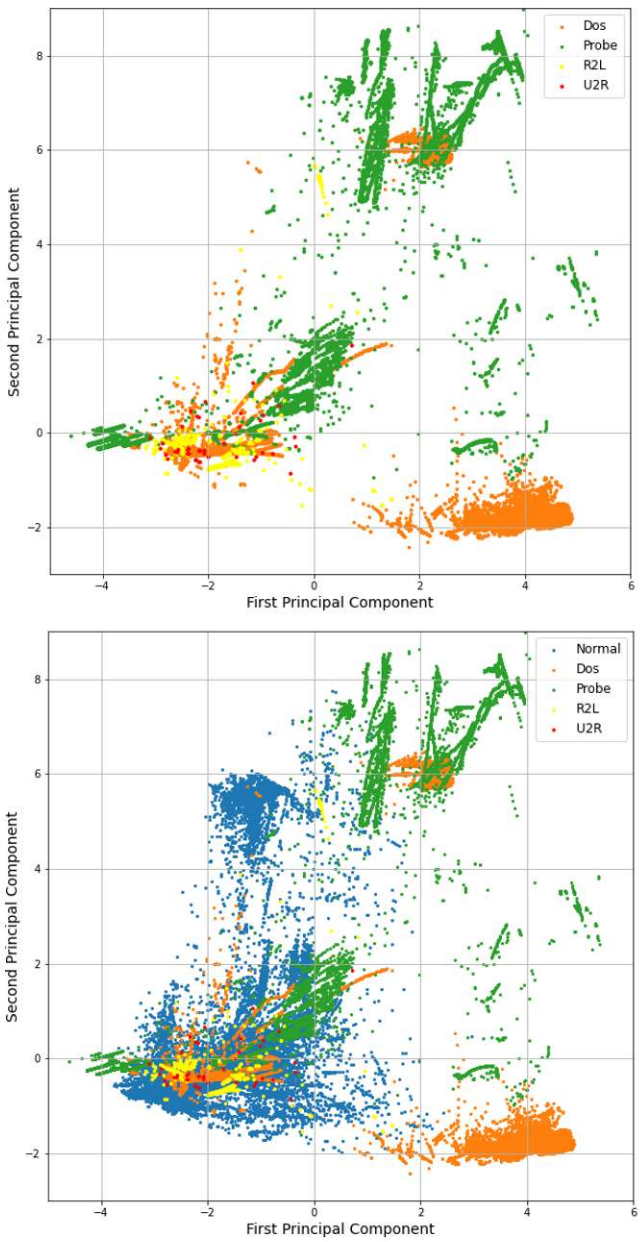


FIGURE 1. 2D-PCA Visualization separated by classes of *KDDTrain+*

detection of R2L and U2R threats, which also led to a high false alarm rate because of their behavioural similarity to normal connections. The information we received from the PCA analysis and previous studies, demonstrates that their models perform well in detecting DOS and Probe but suffer from low detection rates on under-represented attacks. Implying that R2L and U2R attacks need a careful detection strategy. Thus, we designed DLHA in order to address this particular problem.

IV. PROPOSED METHODOLOGY

In this section, we explained the framework overview of our proposed method inspired by the findings of data analysis as displayed in Fig 2. It includes three main steps: data preparation, data transformation, and training and validation processes. Then we demonstrated how DLHA anomaly-based IDS works to detect anomalous connections in a real-time manner. Our approach is also unique in the sense that we first adopted Intersectional Correlated Feature Selection (ICFS), in which intersecting features of different attacks against others are selected. Furthermore, we have two detection layers, where Layer 1 is to detect DOS and Probe attacks out of all connections because of their distinction from others. Then, at Layer 2 we have a dedicated classifier to focus on detecting R2L and U2R threats.

A. A CONCEPTUAL FRAMEWORK OF DLHA

Based on the previous findings, most DOS and Probe attacks significantly deviated from the normal patterns, and R2L and U2R attacks were more similar to normal connections. We designed a conceptual model for a real-time IDS that it should consist of two classifiers. The first classifier needs to be accurate and fast to deal with a large number of network connections simultaneously. The Naive Bayes Classifier is selected based on its efficiency and reliable performance [18], [25]. The second classifier is Support Vector Machine (SVM). It offers a Radial Basis Function (RBF) kernel to solve non-linearly separable problems, which is an effective measure to observe the gap amongst R2L, U2R and normal instances.

1) Data Preparation and Data Transformation

As we have two layers, each layer has its own capability. In order to facilitate this purpose, two groups of data are created based on the original NSL-KDD training data during the data preparation process. The first group contains all instances and classes, while the second group has only R2L, U2R, and Normal instances. At the second step, ICFS, normalization, one-hot encoding, and PCA are implemented. Feature selection technique is a process to select a subset of predictive features and exclude irrelevant features. It not only increases accuracy but also decreases computational time. Nevertheless, feature selection is difficult when the data set contains several classes i.e., the features that are relevant for the specific type of attack might not be predictive for another type of attack. Moreover, it has been proven that different attacks are influenced by different features because the patterns of the attacks vary [1], [78]. For example, TCP protocol is likely to be found in DOS attack [75]. Choosing unimportant features always causes inefficiency in IDS. To handle this problem, we presented ICFS. An example of the ICFS is illustrated in Fig 3.

At this process, we performed feature selection on the two groups using Pearson Correlation Coefficient (PCC). PCC is a bivariate analysis that measures the linear relationship between two random variables, and ranks the features by importance. This method has low computational complexity, and it is scalable for high dimensional data. For numerical features, Pearson's correlation coefficients are used to calculate how much two data points vary together [79]. It is equal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3118573, IEEE Access

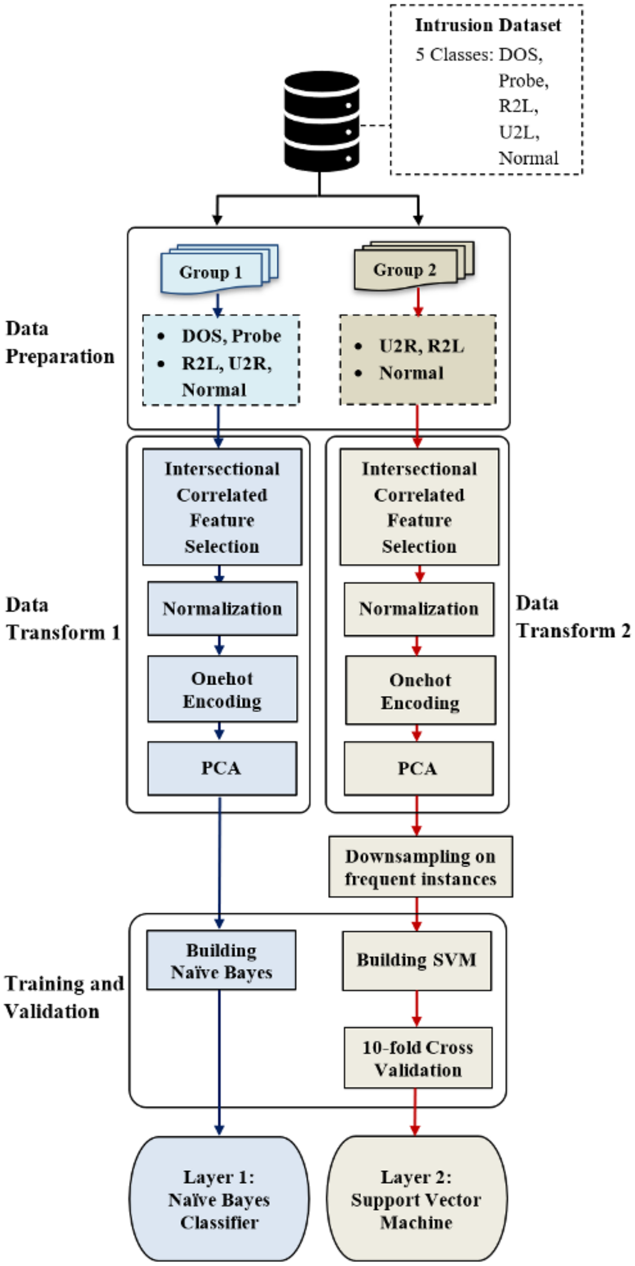


FIGURE 2. A conceptual framework of DLHA anomaly-based IDS

to the covariance divided by the product of their standard deviations. Let X be a random vector with n instances, $X = [x_1, x_2, x_3, \ldots, x_n]$, and Y be a random vector with n instances, $Y = [y_1, y_2, y_3, \ldots, y_n]$, PPC can be expressed as follows:

$$\rho_{x,y} = \frac{\operatorname{cov}(x, y)}{\sigma_x \sigma_y}$$

thus, it can be calculated by

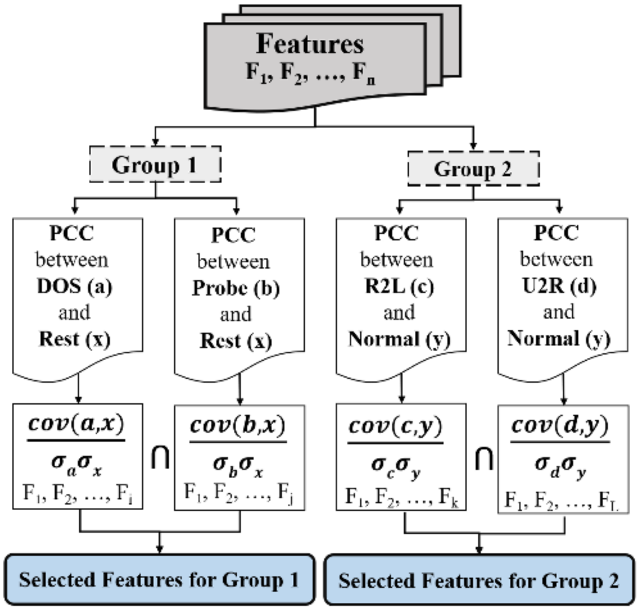

Features
F1, F2, ..., Fn
Group 1
Group 2
PCC between DOS (a) and Rest (x)
PCC between Probe (b) and Rest (x)
PCC between R2L (c) and Normal (y)
PCC between U2R (d) and Normal (y)
cov(a,x) / σaσx
F1, F2, ..., Fi
∩
cov(b,x) / σbσx
F1, F2, ..., Fj
cov(c,y) / σcσy
F1, F2, ..., Fk
∩
cov(d,y) / σdσy
F1, F2, ..., FL
Selected Features for Group 1
Selected Features for Group 2

FIGURE 3. Intersectional Correlated Feature Selection (ICFS)

$$\rho_{x,y} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{3}$$

where n is the number of samples, $\sigma_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$, $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$, and $\operatorname{cov}(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$.

Let F be features $\{F_1, F_2, \ldots, F_n\}$ in training data. In Group 1, we assigned DOS and Probe as 1 and the rest as 0. Let $F_{(DOS)} = \{F_1, F_2, \ldots, F_i\}$ be the features between DOS and the rest, which have PCC greater than 0.1. Let $F_{(Probe)} = \{F_1, F_2, \ldots, F_j\}$ be the features between Probe and the rest, which have PCC greater than 0.1. $F_{(DOS)}$ are predictive features to classify DOS from the rest, $F_{(Probe)}$ are predictive features to classify Probe from the rest. Therefore, $F_{(DOS)} \cap F_{(Probe)}$ are common predictive features to classify DOS and Probe from the rest. As a result, $F_{(DOS)}$ and $F_{(Probe)}$ are the selected features for Group 1. We implemented the same for Group 2 but with a 0.01 threshold because most features are not correlated. In Group 2, R2L and U2R were labelled as 1, and normal records were labelled as 0. Then, PCC was calculated between R2L and Normal as well as U2R and Normal. Consequently, $F_{(R2L)} \cap F_{(U2R)}$ are the selected features for Group 2. The main aim of ICFS is to remove obvious uncorrelated features from the groups. After the ICFS was completed, we normalized the data to be in the range [0,100] as their standard deviations were fairly small. Normalization can be done using a formula in (4).

$$x'_{ij} = \frac{x_{ij} - \min(x)_j}{\max(x)_j - \min(x)_j}, i = 1, 2, \ldots m; j = 1, 2, \ldots, n \tag{4}$$

Afterwards, we performed one-hot encoding and PCA respectively. PCA was used to extract meaningful variance from high dimensional data and turned it into uncorrelated linearly-transformed lower dimensional data. To build an efficient IDS, we only use as few features as possible. Thus, we selected the lowest number that can retain 95% of the variance. We performed data transform individually for each group since the instances are different. This resulted in a difference in the selected features, scaling coefficients, and the number of principal components. Hence, we have two types of data transforms. One-hot encoding and PCA implementation details are presented in Section II. Following the data transform, data balancing in the training set is critical in order to hinder bias towards overwhelming records. Noticeably, we have R2L+U2R = 1,047 instances, and Normal = 67,343 instances, the ratio is approximately 64:1. To prevent bias, downsampling of the majority class is required. For example, 1,047 normal instances were randomly selected in order to make the ratio 1:1. Since the class ratio in Group 1 is not high, the downsampling method was not necessary.

2) Training and Validation

The training and validation steps are vital. Naive Bayes (NB) is selected as a classifier for Group 1. Support Vector Machine (SVM) is selected as a classifier for Group 2.

Naive Bayes Classifier (NBC)

NBC is a simple, yet powerful probabilistic estimator based on applying the Bayes' theorem with an assumption that the considered attributes are independent amongst all. Meaning that each feature influences the result independently [80]. In our proposed method, the NBC's task is to detect DOS and Probe. To serve this goal, DOS and Probe attacks are labelled as 1, and the rest are 0. Let $y = \{y_1, y_2\} = \{$Rest, DOS/Probe$\}$, and let x be a dependent feature vector in the data such that $x = \{x_1, x_2, \ldots, x_n\}$. The Bayes' theorem can be written as follows:

$$P\left(y \mid x_1, \ldots, x_n\right)=\frac{P(y) P\left(x_1, \ldots, x_n \mid y\right)}{P\left(x_1, \ldots, x_n\right)} \tag{5}$$

where $P(y)$ is a prior probability, $P\left(x_1, x_2, \ldots, x_n \mid y\right)$ is the likelihood of a given dependent vector relative to its class, $P(x_1, x_2, \ldots, x_n)$ is a marginal likelihood or evidence. $P\left(y \mid x_1, x_{21} \ldots, x_n\right)$ is the posterior probability of y happening, given $(x_1, x_2, \ldots, x_n)$ has occurred. With the conditional assumption that every feature is independent from each other, it can be defined as:

$$P\left(y \mid x_1, \ldots, x_n\right)=\frac{P(y) \prod_{i=1}^n P\left(x_i \mid y\right)}{P\left(x_1, \ldots, x_n\right)}$$

where n is the number of features after data transform 1. Since $P(x_1, x_2, \ldots, x_n)$ is constant for all. The NBC, then, has the following classification expression:

$$y^{\prime}=\arg \max _y P(y) \prod_{i=1}^n P\left(x_i \mid y\right) \tag{6}$$

As the NBC implements Gaussian algorithm for classification, The $P\left(x_i \mid y\right)$ is assumed to be Gaussian as follows:

$$P\left(x_i \mid y\right)=\frac{1}{\sqrt{2 \pi \sigma_y^2}} \exp \left(-\frac{\left(x_i-\mu_y\right)^2}{2 \sigma_y^2}\right)$$

Despite having the feature-wise independence assumption violated almost all the time in real-world applications, the NBC has demonstrated outstanding classification results in the IDS problem [18]. It is proven to be efficient in detecting frequent DDOS attacks [25]. NBC's computational complexity is defined as $\mathcal{O}(cf)$ where c is the number of classes, and f is the number of features. As the dimensions are reduced in the data transform process, NBC is suitable for dealing with a large amount of connections.

Support Vector Machine (SVM)

SVM is one of the most popular supervised ML algorithm in classification tasks. It was initially proposed in [81], [82] to deal with linear and non-linear optimization problems. SVM creates the best hyperplane in a high-dimensional space in order to separate two classes with the maximum margin between them. It has also been applied to the intrusion detection research area [19], [20], [83]. It provides flexibility in implementations by allowing choices of kernels e.g., linear and radial basis function (RBF). Since RBF is a non-linear support vector classifier (SVC) kernel, it is especially effective in dealing with the data that share complex boundaries [10] i.e., classifying R2L and U2R from normal connections.

For any given training vector pairs of connection-class $\left(x_i, y_i\right), i=1,2, \ldots, n$ where $x_i \in R^n$ and $y \in\{1,-1\}^n$, in which 1 corresponds to a positive class, and -1 corresponds to a negative class. SVM requires a solution to the following problem:

$$\begin{gathered}\min_{w, b, \zeta} \tfrac{1}{2} w^T w+C \textstyle\sum_{i=1}^n \zeta_i \\ \text{subject to } y_i\left(w^T \phi\left(x_i\right)+b\right) \geq 1-\zeta_i \\ \zeta_i \geq 0, i=1, \ldots, n\end{gathered} \tag{7}$$

In the equation, it is attempted to maximize the margin between the two classes by minimizing $w^{\top} w=\|w\|^2$. C is the penalty strength to control misclassified samples at a distance ζ_i from the correct margin boundary that corresponds to the value $y_i\left(w^T \phi\left(x_i\right)+b\right) \geq 1-\zeta_i$. The decision function output for any sample x is defined as:

$$\sum_{i \in S V} y_i \alpha_i K\left(x_i, x\right)+b \tag{8}$$

Its sign is the corresponding class from the prediction. The chosen SVC kernels for validation, in this study, are linear and RBF. Linear kernel is expressed as:

$$K(x, y)=\left(x, x^{\prime}\right)$$

RBF kernel is defined as:

$$K(x, y)=e^{-\gamma\|x-y\|^2}, \quad \gamma>0$$

It has never been confirmed if a non-linear RBF kernel could always perform better than its linear counterpart in this task. Then, we selected linear and RBF as two kernels for the parameter adjustment to observe R2L and U2R boundary. In order to avoid data leakage and data set overfitting, we performed SVM's hyperparameter tuning using 10-fold stratified cross validation within the training set only i.e., *KDDTrain+* and *KDDTrain+_20Percent*. The stratified cross validation is the process of splitting data into folds, in which each fold has to ensure the same proportion of class labels to other folds. The concerned parameters are C and gamma. C is available for both linear and RBF, which is a regularization parameter that adds a penalty for each misclassified instance. The RBF gamma controls the distance of influence of a single training sample. The set of parameters are as follows; linear: C = 0.01, 0.1, 1, 10, 100, 1000, RBF: C = 0.1, 1, 10, 100 and gamma = 0.01, 0.1, 1, 10. Consequently, we have six parameters for the linear kernel and 16 parameter sets for the RBF kernel.

SVM was implemented by using LIBSVM [84]. The machine specification is on Ubuntu 20.04 LTS, Intel Corei9-9900 3.10GHz, and 32GB of RAM.

B. DLHA ALGORITHM

Real-time traffic classification using DLHA is displayed in Fig 4. DLHA is proposed to improve the overall detection rate, and especially the detection rate of rare attacks that are more hostile i.e., R2L and U2R in this study. It is also designed to be an efficient real-time IDS since we have ICFS and PCA to reduce data dimensions as much as possible. DLHA algorithm works as follows: the network connection packages are captured and sent through Data Transformation 1 process, then the transformed data are passed to Layer 1, which is NBC, to determine if the connection is DOS, Probe, or Normal. If the prediction is negative, then the connection is highly unlikely to be DOS or Probe. Then, the second layer is activated. The original data are sent through Data Transformation 2 process. Then the transformed data are passed to Layer 2, which is SVM, to determine if the connection is R2L, U2R or normal. If the prediction is negative, this connection is expected to be normal. If any of the two classifiers predicted positive, the connection is terminated and marked as an anomaly. Since DOS and Probe attacks are more likely to occur, this framework is computationally efficient to detect DOS and Probe first, then R2L and U2R subsequently. DLHA algorithm is explained as follows:

Algorithm 1 DLHA algorithm

```
Input: $X = \{f_1, f_2, \ldots, f_{40}\}$ // 40 attributes captured
Output: $y \in \{0, 1\}$
  while DLHA IDS is running do
    // for every network connection
    after performing data transform 1
    represent $X_i$ as $X_{t_1}$
    if $Layer1$ predicts $X_{t_1}$ as 1 then
      $y \leftarrow 1$
      return $y$
    else
      Layer 2 is activated
      after performing data transform 2
      represent $X_i$ as $X_{t_2}$
      if $Layer2$ predicts $X_{t_2}$ as 1 then
        $y \leftarrow 1$
        return $y$
      else
        $y \leftarrow 0$
        return $y$
      end if
    end if
  end while
```

As our 2-classifier hybrid approach is dedicated to maximizing the detection rates of R2L and U2R attacks, there are few continuing costs of operation as a trade-off. Firstly, time spent on attack detection increases because the decision process becomes more complex, where two negative predictions are required to confirm that the connection is safe. Additionally, performing data transformation for each layer leads to higher resource consumption. Powerful machines are recommended for this approach to avoid traffic bottlenecks. Significantly, machine learning approaches rely on quality data to establish a reliable model. Collecting attack signatures e.g., using a honeypot strategy, would be beneficial for a long term IDS implementation [62].

V. EVALUATION AND RESULT

To evaluate the performance of our proposed DLHA, we conducted experiments using the two training data sets *KDDTrain+* and *KDDTrain+_20Percent* in order to analyze the framework on a large sample size and a small sample size. To measure generalization of the model, training and validation were only implemented using training data as described in Section IV. Thus, the testing data in *KDDTest+* are left unseen.

A. EVALUATION METRICS

There are five metrics presented in this work i.e., 1) Accuracy, 2) F1 Score, 3) Precision, 4) Detection Rate (Recall), and 5) False Alarm Rate. The four measures used to calculate the metrics are presented as follows: True Positive (TP) = correctly predicted attacks, True Negative (TN) = correctly predicted normal instances, False Positive (FP) = incorrectly predicted attacks, and False Negative (FN) = incorrectly predicted normal instances.

1. Accuracy is the overall percentage of correct classification. However, it is unreliable for imbalanced data set, particularly for the IDS problem. It can be computed as:

$$\frac{(TP + TN)}{TP + TN + FP + FN} \tag{9}$$

2. F1 Score is the harmonic mean of precision and recall.

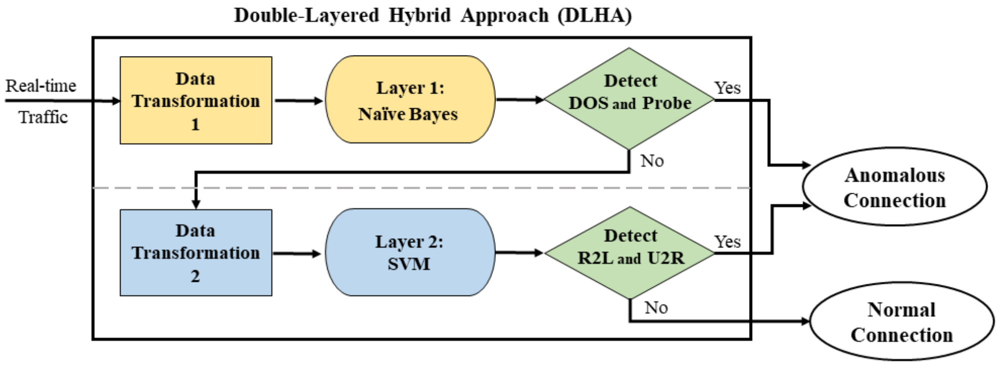


FIGURE 4. Real-time traffic classification using DHLA

It can be computed as:

$$2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{10}$$

3. Precision is the classification ability to correctly detect attacks out of the total positive predictions. It can be computed as:

$$\frac{TP}{TP + FP} \tag{11}$$

4. Detection Rate (Recall) is the classification ability to correctly predict attacks from actual attacks. It can be computed as:

$$\frac{TP}{TP + FN} \tag{12}$$

5. False Alarm Rate is the proportion of wrongly predicting attacks. FAR infers overestimation that falsely requires human interference. It can be computed as:

$$\frac{FP}{FP + TN} \tag{13}$$

In this work, we mainly focused on Detection Rate (DR). DR is critical because it implies how many attacks the model can identify out of the total number of actual attacks.

B. EXPERIMENTAL RESULT

At the training stage, we re-created the training data into 2 groups as mentioned previously. Then, we conducted the ICFS. The correlated features between DOS and the rest $\{F_1, F_2, \ldots, F_i\}$ are [8, 12, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41] (see Table 5). The correlated features between Probe and the rest $\{F_1, F_2, \ldots, F_j\}$ = [1, 12, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39, 40, 41]. Therefore, the intersect features of DOS/Probe are [12, 23, 25, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39, 40, 41] In group 2, the correlated features between R2L and Normal $\{F_1, F_2, \ldots, F_k\}$ are [1, 5, 6, 9, 10, 11, 12, 14, 18, 22, 23, 24, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]. The correlated features between U2R and Normal $\{F_1, F_2, \ldots, F_l\}$ are [9, 10, 12, 14, 17, 18, 24, 31, 32, 33, 36, 37]. Hence, the intersect features of R2L/U2R are [9, 10, 12, 14, 18, 24, 31, 32, 33, 36, 37]. This is reasonable e.g., *count* is commonly high in DOS and Probe attacks, and *num_shells* is commonly relevant to R2L and U2R patterns.

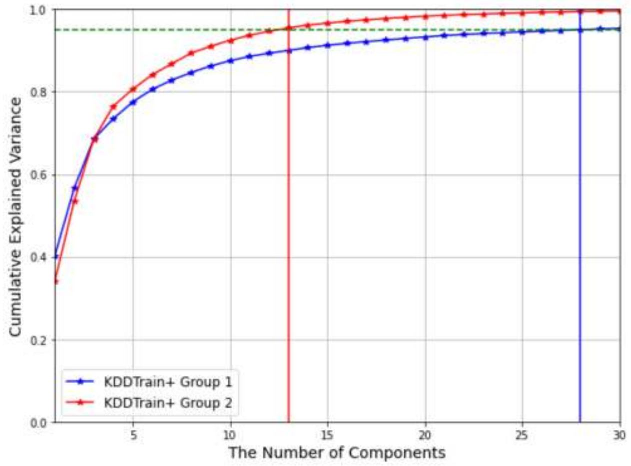


FIGURE 5. Cumulative explained variance against the number of principal components measured in both groups to select the optimal number of dimensions

After that, normalization and one-hot encoding were performed respectively. PCA is the last step in the Data Transform process. 95% of cumulative variance was chosen as a threshold. The cumulative variance against the number of principal components is visualized in Fig 5. It indicated that 28 is the suitable number of components in Group 1, which represented 95.07% of variance. 13 is the selected number of components in Group 2, which constituted 96.55% variance.

Then, downsampling was carried out on the frequent records i.e., Normal on Group 2 to keep a 1:1 ratio between anomaly and normal. At the last step, we performed hyperparameter tuning on Group 2 with a series of linear and RBF kernel parameters. The same set of parameters was also implemented on the comparatively smaller data set i.e., *KDDTrain+_20Percent* to evaluate a variety of different configurations with a primary performance boost based on the stratified 10-fold cross validation method. The results were shown in Fig 6, and Fig 7 respectively. Our main goal is to maximize the detection rates of the model in order to prevent losses caused by intruders. Accordingly, each box-and-whisker plot measured the detection rates as a result of each testing fold from 10 folds. The horizontal line in the box indicated the median detection rate value, and the + specified the average detection rate of 10 scores.

The first experiment used *KDDTrain+* in training. We attempted to select the best parameters to classify R2L and U2R attacks out of normal instances. Fig 6 indicated that linear kernel performed well on lower C and dropped its performance on higher C. The RBF kernel performed comparatively better in most combinations of parameters. There is an exception that when C is equal to 0.1 and gamma is equal to 10, where the SVM performance is significantly lowered. It is evident that the higher the gamma value is, when C is equal to 0.1, the more the detection rate dropped. Additionally, when C is equal to or greater than 1, the performances are relatively consistent as seen in configurations 10-21. The highest detection rate is located at configuration 6, where C equals 0.1 and gamma equals 0.01. It accomplished an acceptable average detection rate of 0.9943 with STD = 0.0061 and 0.1337 in FAR.

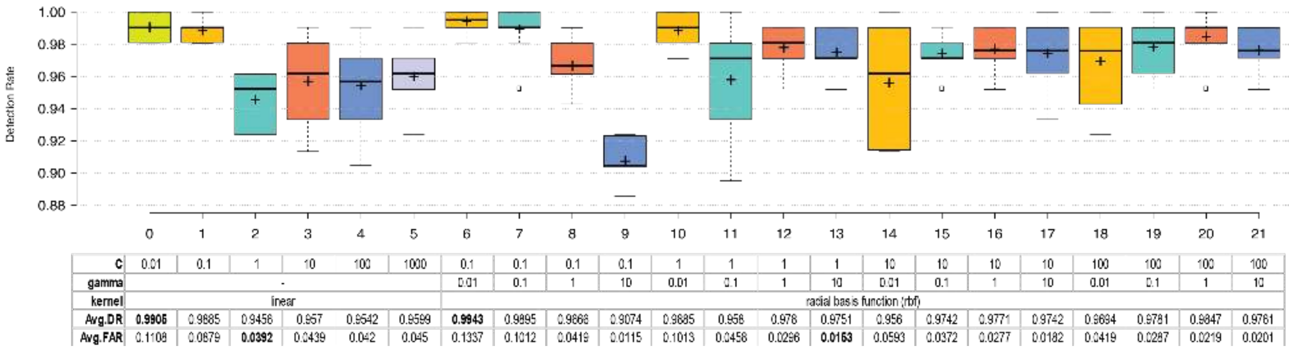
The second experiment used *KDDTrain+_20Percent* in training. In Fig 7, we observed a small difference where the configurations in linear kernel performed moderately better compared to its previous evaluation. Most configurations of linear kernel performed worse when the data set becomes larger as shown in Fig 6. Noticeably, the same pattern is confirmed in a smaller data set, that the RBF kernel has a similar performance in configurations no.10-21. It performed best when C is equal to 0.1, and gamma is approximately 0.01 or 0.1. The performance dramatically dropped when C is equal to 0.1 and gamma is equal to 10 by reducing to lower than 0.6 in the detection rates on some testing folds. The highest detection rate is attained in configuration 7, where C equals 0.1, and gamma equals 0.1 by acquiring the average detection rate of 0.9864 with STD = 0.0291 and 0.1136 in FAR. The results intuitively suggested that in order to detect R2L and U2R accurately, the penalty on misclassified samples should not be high (low C), and a single training instance should not have too much influence on the decision boundary (low gamma).

To evaluate our framework on the two experiments, we tested DLHA on the unseen data i.e., *KDDTest+* using the procedure explained in Algorithm 1 and the best parameters derived from the CV process. Our proposed framework presented outstanding classification results achieving 88.97% in accuracy, 90.57% in F1 score, 88.17% in precision, and 93.11% in detection rate with 11.82% of false alarm rate by using *KDDTrain+* in training. The framework was also proven effective in a comparatively smaller data set i.e., using only 20% of all samples (*KDDTrain+_20Percent*) in training, where it obtained acceptable results, these being 87.55% accuracy, 89.19% in F1 score, 88.17% in precision, and 90.24% in detection rate with 11.83% of false alarm rate.

Then, we conducted a detailed analysis of our results to explore the detection rates of each class as shown in Fig 8. It was found that our proposed method, from using *KDDTrain+* in training, has the detection rates of 92.4% on DOS (6,893 out of 7460), 90.87% on Probe (2,200 out of 2,421), 96.67% on R2L (2,789 out of 2,885), and 100% on U2R (67 out of 67). When using *KDDTrain+_20Percent* in training, it has the detection rates of 92.84% on DOS (6,926 out of 7,460), 89.88% on Probe (2,176 out of 2,421), 83.6% on R2L (2,421 out of 2,885), and 100% on U2R (67 out of 67). Therefore, it is demonstrated that our proposed DLHA accomplished its objective in maintaining great detection rates on DOS and Probe, and showed excellent performance in detecting 96.67% on R2L and 100% on U2R in *KDDTest+*. In addition, the time measurement was also presented as displayed in Fig 9. The presented numbers were the average of 10 times running on the desktop machine. It was apparent that the time used for training in the *KDDTrain+_20Percent* was only one-third of the full data set as it contains only 20% of all training data. The testing time is similar on both training sets, where approximately 2.5 seconds were spent classifying 22,544 instances, or in other words, that ≈ 9,000 instances were successfully classified in one second.

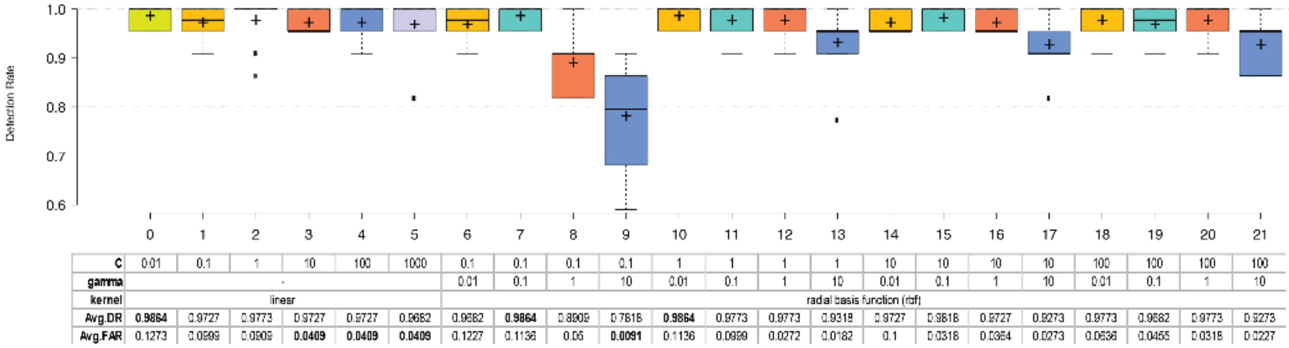
One of the most important areas we highlighted in this study is how successful our approach is in detecting additional attack categories in *KDDTest+*, the attack categories that are absent in the training data set. There are 12,833 attacks in *KDDTest+*, 9,083 belong to known attack categories, and 3,750 are in unseen attack categories. DLHA, using *KDDTrain+* in training, achieved detection rates of 94.01% (8,539 out of 9,083) from known attack categories, and 90.90% (3,411 out of 3,750) from unseen attack categories. DLHA, using *KDDTrain+_20Percent* in training, achieved detection rates of 89.81% (8,157 out of 9,083) from known attack categories, and 91.28% (3,423 out of 3,750) from unseen attack categories. From the results, DLHA performed outstandingly well in detecting both known and unknown attack categories. DLHA trained on *KDDTrain+_20Percent* gained a slightly higher detection rate on unseen attack categories. However, DLHA detected 94.01% of attacks from known attack categories when the total samples were used in training due to a greater amount of the samples per each category in *KDDTrain+* compared to *KDDTrain+_20Percent*.

It is worth mentioning that there are a number of existing works that previously studied anomaly-based IDS using a refined version of the KDD99 i.e., NSL-KDD [1], the same data set we considered in this study. However, some scholars presented their results from implementing a cross validation method, a holdout method, or using a portion of the KDD99 data set, which are not sufficiently reliable in the context of IDS research i.e., achieved over 99-100% in accuracy or detection rate [28]. In this study, we used *KDDTrain+* and *KDDTrain+_20Percent* in the training and validation steps, and only used *KDDTest+* in testing. Therefore, we only compared our results to the studies that take a similar



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
C	0.01	0.1	1	10	100	1000	0.1	0.1	0.1	0.1	1	1	1	1	10	10	10	10	100	100	100	100
gamma	-						0.01	0.1	1	10	0.01	0.1	1	10	0.01	0.1	1	10	0.01	0.1	1	10
kernel	linear						radial basis function (rbf)															
Avg.DR	**0.9905**	0.9885	0.9456	0.957	0.9542	0.9599	**0.9943**	0.9895	0.9868	0.9074	0.9885	0.958	0.978	0.9751	0.956	0.9742	0.9771	0.9742	0.9694	0.9781	0.9847	0.9761
Avg.FAR	0.1108	0.0879	**0.0392**	0.0439	0.042	0.045	0.1337	0.1012	0.0419	0.0115	0.1013	0.0458	0.0296	**0.0153**	0.0593	0.0372	0.0277	0.0182	0.0419	0.0287	0.0219	0.0201

FIGURE 6. Box-and-whisker plots present mean, median, range, and quartile distribution of the detection rates from different parameters for SVM 10-fold CV in *KDDTrain+*



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
C	0.01	0.1	1	10	100	1000	0.1	0.1	0.1	0.1	1	1	1	1	10	10	10	10	100	100	100	100
gamma	-						0.01	0.1	1	10	0.01	0.1	1	10	0.01	0.1	1	10	0.01	0.1	1	10
kernel	linear						radial basis function (rbf)															
Avg.DR	**0.9864**	0.9727	0.9773	0.9727	0.9727	0.9682	0.9682	**0.9864**	0.8909	0.7818	**0.9864**	0.9773	0.9773	0.9318	0.9727	0.9818	0.9727	0.9273	0.9773	0.9682	0.9773	0.9273
Avg.FAR	0.1273	0.0909	0.0909	**0.0409**	**0.0409**	**0.0409**	0.1227	0.1136	0.05	**0.0091**	0.1136	0.0909	0.0272	0.0182	0.1	0.0318	0.0364	0.0273	0.0636	0.0455	0.0318	0.0227

FIGURE 7. Box-and-whisker plots present mean, median, range, and quartile distribution of the detection rates from different parameters for SVM 10-fold CV in *KDDTrain+_20Percent*

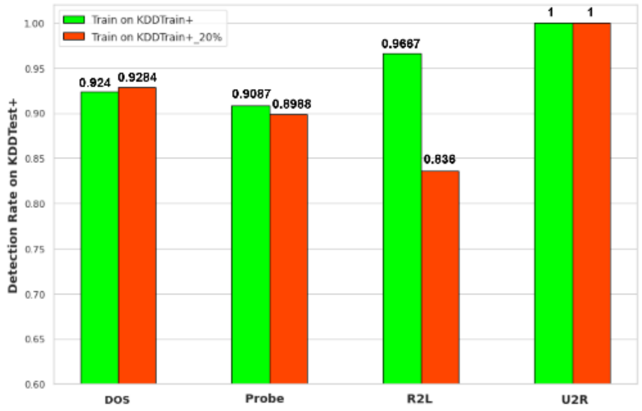


FIGURE 8. Detection rates of major attack categories

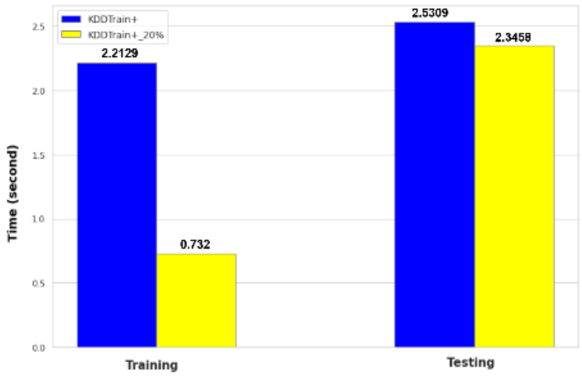


FIGURE 9. Training and testing time of DLHA in seconds

approach i.e., using the *KDDTest+* in testing.

In order to objectively evaluate our proposed framework on wider impacts, we conducted an extensive comparison of our results to other publicly published IDS research papers as shown in Table 6. It is acknowledged that our framework is highly competitive in the field. Evidently, DLHA obtains the highest F1 Score and DR. However, the obvious downside of our model is a relatively high FAR because we attempt to maximize the detection rate. The no.22-26 results are derived from the original NSL-KDD article, which are set as a baseline. Any models that perform worse than the baseline are considered substandard. Our DLHA has considerably higher accuracy than the best baseline single machine learning classifier, NB Tree, by +6.95%, and +11.56% compared to Multi-Layer Perceptron. Furthermore, [37], [47] developed the single machine learning classifier models, SVM and RNN, to detect all attack types. Their accuracy scores were 82.37% and 81.29% respectively, indicating no improvement over the baseline, while most hybrid methods performed better than the baseline. In addition, we compared our detection rates of the major attack categories to other studies as displayed in

TABLE 6. Performance comparison on Accuracy, F1 Score, Precision, Detection Rate, and False Alarm Rate with other anomaly-based IDS approaches (only compared to the studies that performed evaluation in the original *KDDTest+*)

No.	Algorithm	Year	Training set	Accuracy(%)	F1 Score(%)	Precision(%)	Detection Rate(%)	FAR(%)
1	Our proposed DLHA	2021	KDDTrain+	88.97	**90.57**	88.17	**93.11**	11.82
2	Our proposed DLHA	2021	KDDTrain+_20%	87.55	89.19	88.16	90.24	11.83
3	Hybrid K-Means+RF [69]	2021	KDDTrain+	85.24	-	-	-	-
4	CNN+BiLSTM [34]	2020	KDDTrain+	83.58	85.14	85.82	84.49	-
5	BAT-MC [73]	2020	KDDTrain+	84.25	-	-	-	-
6	Autoencoder [65]	2020	KDDTrain+	84.24	81.98	87	80.37	-
7	CBR-CNN [74]	2019	80% KDDTrain+	**89.41**	-	-	-	-
8	TSE-IDS [28]	2019	KDDTrain+	85.797	-	88.0	86.8	11.7
9	Adaptive Ensemble [11]	2019	KDDTrain+	85.2	84.9	86.5	86.5	-
10	Multi Tree algorithm [11]	2019	KDDTrain+	84.23	-	-	-	-
11	Sparse AE + SVM [85]	2018	KDDTrain+	84.96	85.28	**96.23**	76.57	-
12	MOPF [67]	2017	KDDTrain+	84.12	-	-	-	-
13	RNN-IDS [37]	2017	KDDTrain+	81.29	-	-	-	-
14	TDTC [4]	2016	KDDTrain+	-	-	-	84.86	4.86
15	TDTC [4]	2016	KDDTrain+_20%	-	-	-	84.82	5.56
16	PSOM+PCA+FDR [63]	2015	KDDTrain+	88.0	-	-	92.0	-
17	Two-Tier [60]	2015	KDDTrain+	-	-	-	81.97	5.44
18	Two-Tier [60]	2015	KDDTrain+_20%	-	-	-	83.24	4.83
19	GAR-Forest [86]	2015	KDDTrain+	85.06	85.1	87.5	85.1	12.2
20	SVM-IDS [47]	2014	KDDTrain+	82.37	-	-	-	15.0
21	Fuzzy with Evolution [87]	2011	KDDTrain+	82.74	-	-	-	**3.9**
22	NB Tree [3]	2009	KDDTrain+_20%	82.02	-	-	-	-
23	Random Tree [3]	2009	KDDTrain+_20%	81.59	-	-	-	-
24	J48 DT Learning [3]	2009	KDDTrain+_20%	81.05	-	-	-	-
25	Random Forest [3]	2009	KDDTrain+_20%	80.67	-	-	-	-
26	Multi-Layer Perceptron [3]	2009	KDDTrain+_20%	77.41	-	-	-	-

TABLE 7. Comparative detection rates of major attack categories

Algorithm	DOS	Probe	R2L	U2R
Our proposed DLHA	92.4	90.87	**96.67**	**100**
TDTC [60]	88.20	87.32	42	70.15
Hybrid K-Means+RF [69]	90.42	91.53	73.84	25.79
Two-tier [4]	84.68	79.76	34.81	67.16
PCA+KNN [88]	94.23	78.86	69.87	80.09
HFR-MLR [43]	89.7	80.2	34.2	29.5
SVM+BIRCH [89]	**97.5**	**99.5**	19.7	28.8
Two-Level [90]	97.37	94.72	14.02	90.71
Siam-IDS [91]	85.37	48.66	33.25	56.72
Adaptive Ensemble [11]	84.37	87.11	55.27	25.0

Table 7. The comparison indicates that DLHA is not the best algorithm to detect DOS or Probe, as our results attain approximately 90-92%, while others show superior outcomes. However, our model can accurately detect every type of attack compared to others that exhibit undesirable detection scores on R2L and U2R. Our model clearly outperforms all other methods by reaching the detection rates of 96.67% in R2L and 100% in U2R.

VI. CONCLUSION

Rule-based IDS methods are not sufficient for the new era of rapidly-growing internet connections worldwide. Anomaly-based IDS approaches using machine learning offer a promising performance, but usually suffer from bias towards frequent attacks as well as underestimation of rare threats. Single machine learning models are not accurate in detecting all types of attacks, which result in a low detection rate, particularly on infrequent attacks. Thus, the IDS problem requires a hybrid solution.

This paper proposed an algorithm called a Double-Layered Hybrid Approach (DLHA) to tackle an unsatisfactory performance on rare attacks, which also give rise to an improved overall detection rate. An Intersectional Correlated Feature Selected (ICFS) was presented as part of DLHA to exclude commonly irrelevant features on the subgroups to reduce dimensions and accelerate the whole framework for real-time practice. The detection part consists of two layers. The first layer utilized NBC to classify DOS and Probe attacks from all connections. The second layer adopted SVM with RBF kernel to detect R2L and U2R attacks among normal traffic, which is a more difficult task. Hyperparameter tuning is paramount, c and gamma on SVM were optimized as they are the primary factors to accurately detect attacks that share a similar pattern to normal connections i.e., R2L and U2R. Our proposed DLHA was evaluated on the NSL-KDD data set. It achieved exceptional results with an overall detection rate of 93.11% with over 96.67% detection rate of R2L, and 100% of U2R. The execution time and F1 score have proven its enhanced efficiency and capability for broader applications.

Our experimental results demonstrated how successful and effective the hybrid IDS approach is by using two different classifiers with ICFS. Since we avoided overfitting and data leakage by implementing hyperparameter tuning on 10-fold CV using training data, we concluded that our DLHA offers a generalized model with a class-topping performance in detecting uncommon but more dangerous attacks. This approach is suitable for a real-time IDS and aims to secure critical network environments.

REFERENCES

[1] U. S. Musa, M. Chhabra, A. Ali, and M. Kaur, "Intrusion detection system using machine learning techniques: A review," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 2020, pp. 149–155.

[2] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[3] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. IEEE, 2009, pp. 1–6.

[4] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 2, pp. 314–323, 2016.

[5] R. Das and M. Z. Gündüz, "Analysis of cyber-attacks in iot-based critical infrastructures," *International Journal of Information Security Science*, vol. 8, no. 4, pp. 122–133, 2020.

[6] M. Roesch *et al.*, "Snort: Lightweight intrusion detection for networks." in *Lisa*, vol. 99, no. 1, 1999, pp. 229–238.

[7] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, "Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, vol. 2, 2000, pp. 12–26 vol.2.

[8] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[9] F. Salo, M. Injadat, A. B. Nassif, A. Shami, and A. Essex, "Data mining techniques in intrusion detection systems: A systematic literature review," *IEEE Access*, vol. 6, pp. 56 046–56 058, 2018.

[10] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE access*, vol. 6, pp. 33 789–33 795, 2018.

[11] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82 512–82 521, 2019.

[12] J. Gao, S. Chai, B. Zhang, and Y. Xia, "Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis," *Energies*, vol. 12, no. 7, p. 1223, 2019.

[13] S.-H. Kang and K. J. Kim, "A feature selection approach to find optimal feature subsets for the network intrusion detection system," *Cluster Computing*, vol. 19, no. 1, pp. 325–333, 2016.

[14] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, 2008.

[15] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.

[16] J. Mill and A. Inoue, "Support vector classifiers and network intrusion detection," in *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No. 04CH37542)*, vol. 1. IEEE, 2004, pp. 407–410.

[17] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection system based on knn classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, vol. 2014, 2014.

[18] B. Zhang, Z. Liu, Y. Jia, J. Ren, and X. Zhao, "Network intrusion detection method based on pca and bayes algorithm," *Security and Communication Networks*, vol. 2018, 2018.

[19] P. Tao, Z. Sun, and Z. Sun, "An improved intrusion detection algorithm based on ga and svm," *Ieee Access*, vol. 6, pp. 13 624–13 631, 2018.

[20] Y. Zhang, Q. Yang, S. Lambotharan, K. Kyriakopoulos, I. Ghafir, and B. AsSadhan, "Anomaly-based network intrusion detection using svm," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2019, pp. 1–6.

[21] M. Li, "Application of cart decision tree combined with pca algorithm in intrusion detection," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2017, pp. 38–41.

[22] S. Sahu and B. M. Mehtre, "Network intrusion detection system using j48 decision tree," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 2023–2026.

[23] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & security*, vol. 21, no. 5, pp. 439–448, 2002.

[24] M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," *International journal of computer science and network security*, vol. 7, no. 12, pp. 258–263, 2007.

[25] R. F. Fouladi, C. E. Kayatas, and E. Anarim, "Frequency based ddos attack detection approach using naive bayes classification," in *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, 2016, pp. 104–107.

[26] S. Mukherjee and N. Sharma, "Intrusion detection using naive bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, 2012.

[27] J. V. Anand Sukumar, I. Pranav, M. Neetish, and J. Narayanan, "Network intrusion detection using improved genetic k-means algorithm," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018, pp. 2441–2446.

[28] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "Tse-ids: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94 497–94 507, 2019.

[29] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 science and information conference*. IEEE, 2014, pp. 372–378.

[30] W. Feng, Q. Zhang, G. Hu, and J. X. Huang, "Mining network data for intrusion detection through combining svms with ant colony networks," *Future Generation Computer Systems*, vol. 37, pp. 127–140, 2014.

[31] M. Ektefa, S. Memar, F. Sidi, and L. S. Affendey, "Intrusion detection using data mining techniques," in *2010 International Conference on Information Retrieval & Knowledge Management (CAMP)*. IEEE, 2010, pp. 200–203.

[32] H. F. Eid, A. E. Hassanien, T.-h. Kim, and S. Banerjee, "Linear correlation-based feature selection for network intrusion detection model," in *International Conference on Security of Information and Communication Networks*. Springer, 2013, pp. 240–248.

[33] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *Journal of information security and applications*, vol. 44, pp. 80–88, 2019.

[34] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, vol. 8, pp. 32 464–32 476, 2020.

[35] L. Mohammadpour, T. C. Ling, C. S. Liew, and C. Y. Chong, "A convolutional neural network for network intrusion detection system," *Proceedings of the Asia-Pacific Advanced Network*, vol. 46, pp. 50–55, 2018.

[36] Y. Ding and Y. Zhai, "Intrusion detection system for nsl-kdd dataset using convolutional neural networks," in *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, 2018, pp. 81–85.

[37] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *Ieee Access*, vol. 5, pp. 21 954–21 961, 2017.

[38] P. Gogoi, D. Bhattacharyya, B. Borah, and J. K. Kalita, "Mlh-ids: a multi-level hybrid intrusion detection method," *The Computer Journal*, vol. 57, no. 4, pp. 602–623, 2014.

[39] H. Yao, Q. Wang, L. Wang, P. Zhang, M. Li, and Y. Liu, "An intrusion detection framework based on hybrid multi-level data mining," *International Journal of Parallel Programming*, vol. 47, no. 4, pp. 740–758, 2019.

[40] N. B. Nanda and A. Parikh, "Hybrid approach for network intrusion detection system using random forest classifier and rough set theory for rules generation," in *International Conference on Advanced Informatics for Computing Research*. Springer, 2019, pp. 274–287.

[41] P. Singh and M. Venkatesan, "Hybrid approach for intrusion detection system," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*. IEEE, 2018, pp. 1–5.

[42] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on abc-afs algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018.

[43] E. Kim and S. Kim, "A novel anomaly detection system based on hfr-mlr method," in *Mobile, Ubiquitous, and Intelligent Computing*. Springer, 2014, pp. 279–286.

[44] P.-J. Chuang and S.-H. Li, "Network intrusion detection using hybrid machine learning," in *2019 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*. IEEE, 2019, pp. 1–5.

[45] J. Esmaily, R. Moradinezhad, and J. Ghasemi, "Intrusion detection system based on multi-layer perceptron neural networks and decision tree," in *2015 7th Conference on Information and Knowledge Technology (IKT)*. IEEE, 2015, pp. 1–5.

[46] L. Li, Y. Yu, S. Bai, Y. Hou, and X. Chen, "An effective two-step intrusion detection approach based on binary classification and k-nn," *IEEE Access*, vol. 6, pp. 12 060–12 073, 2017.

[47] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in nsl-kdd cup 99 dataset employing svms," in *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*. IEEE, 2014, pp. 1–6.

[48] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[49] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *Eai Endorsed Transactions on Security and Safety*, vol. 3, no. 9, p. e2, 2016.

[50] A. A. Olusola, A. S. Oladele, and D. O. Abosede, "Analysis of kdd'99 intrusion detection dataset for selection of relevance features," in *Proceedings of the world congress on engineering and computer science*, vol. 1. WCECS, 2010, pp. 20–22.

[51] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class svm," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017.

[52] G. Li, Z. Yan, Y. Fu, and H. Chen, "Data fusion for network intrusion detection: a review," *Security and Communication Networks*, vol. 2018, 2018.

[53] B. Aslahi-Shahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami, M. J. Golkar, and A. Ebrahimi, "A hybrid method consisting of ga and svm for intrusion detection system," *Neural computing and applications*, vol. 27, no. 6, pp. 1669–1676, 2016.

[54] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[55] N. A. Biswas, F. M. Shah, W. M. Tammi, and S. Chakraborty, "Fp-ank: An improvised intrusion detection system with hybridization of neural network and k-means clustering over feature selection by pca," in *2015 18th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2015, pp. 317–322.

[56] M. Mazini, B. Shirazi, and I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and adaboost algorithms," *Journal of King Saud University-Computer and Information Sciences*, vol. 31, no. 4, pp. 541–553, 2019.

[57] H. Altwaijry, "Bayesian based intrusion detection system," in *IAENG Transactions on Engineering Technologies*. Springer, 2013, pp. 29–44.

[58] Ü. Çavuşoğlu, "A new hybrid approach for intrusion detection using machine learning methods," *Applied Intelligence*, vol. 49, no. 7, pp. 2735–2761, 2019.

[59] T. S. Hwang, T.-J. Lee, and Y.-J. Lee, "A three-tier ids via data mining approach," in *Proceedings of the 3rd annual ACM workshop on Mining network data*, 2007, pp. 1–6.

[60] H. H. Pajouh, G. Dastghaibyfard, and S. Hashemi, "Two-tier network anomaly detection model: a machine learning approach," *Journal of Intelligent Information Systems*, vol. 48, no. 1, pp. 61–74, 2017.

[61] A. A. Alfantookh, "Dos attacks intelligent detection using neural networks," *Journal of King Saud University-Computer and Information Sciences*, vol. 18, pp. 31–51, 2006.

[62] M. Baykara and R. Das, "A novel honeypot based security approach for real-time intrusion detection and prevention systems," *Journal of Information Security and Applications*, vol. 41, pp. 103–116, 2018.

[63] E. De la Hoz, E. De La Hoz, A. Ortiz, J. Ortega, and B. Prieto, "Pca filtering and probabilistic som for network intrusion detection," *Neurocomputing*, vol. 164, pp. 71–81, 2015.

[64] A. Karami and M. Guerrero-Zapata, "A fuzzy anomaly detection system based on hybrid pso-kmeans algorithm in content-centric networks," *Neurocomputing*, vol. 149, pp. 1253–1269, 2015.

[65] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, 2020.

[66] M. Baykara and R. Das, "Softswitch: a centralized honeypot-based security approach using software-defined switching for secure management of vlan networks," *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, vol. 27, pp. 3309–3325, 09 2019.

[67] H. Bostani and M. Sheikhan, "Modification of supervised opf-based intrusion detection systems using unsupervised learning and social network concept," *Pattern Recognition*, vol. 62, pp. 56–72, 2017.

[68] A. Golrang, A. M. Golrang, S. Yildirim Yayilgan, and O. Elezaj, "A novel hybrid ids based on modified nsgaii-ann and random forest," *Electronics*, vol. 9, no. 4, p. 577, 2020.

[69] C. Liu, Z. Gu, and J. Wang, "A hybrid intrusion detection system based on scalable k-means+ random forest and deep learning," *IEEE Access*, vol. 9, pp. 75 729–75 740, 2021.

[70] S. Stolfo *et al.*, "Kdd-99 dataset," *Available on http://www. kdd. ics. uci. edu/databases/kddcup99/kddcup99. html kddcup99. html*, 1999.

[71] A. Özgür and H. Erdem, "A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *PeerJ Preprints*, vol. 4, p. e1954v1, 2016.

[72] S. Revathi and A. Malathi, "A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 12, pp. 1848–1853, 2013.

[73] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "Bat: Deep learning methods on network intrusion detection using nsl-kdd dataset," *IEEE Access*, vol. 8, pp. 29 575–29 585, 2020.

[74] N. Chouhan, A. Khan *et al.*, "Network anomaly detection using channel boosted and residual learning based deep convolutional neural network," *Applied Soft Computing*, vol. 83, p. 105612, 2019.

[75] L. Dhanabal and S. Shantharajah, "A study on nsl-kdd dataset for intrusion detection system based on classification algorithms," *International journal of advanced research in computer and communication engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[76] C. Seger, "An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing," 2018.

[77] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[78] N. A. Biswas, F. M. Shah, W. M. Tammi, and S. Chakraborty, "Fp-ank: An improvised intrusion detection system with hybridization of neural network and k-means clustering over feature selection by pca," in *2015 18th International Conference on Computer and Information Technology (ICCIT)*, 2015, pp. 317–322.

[79] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[80] H. Zhang, "The optimality of naive bayes," *AA*, vol. 1, no. 2, p. 3, 2004.

[81] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.

[82] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[83] J. Jha and L. Ragha, "Intrusion detection system using support vector machine," *International Journal of Applied Information Systems (IJAIS)*, vol. 3, pp. 25–30, 2013.

[84] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[85] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with svm for network intrusion detection," *IEEE Access*, vol. 6, pp. 52 843–52 856, 2018.

[86] N. K. Kanakarajan and K. Muniasamy, "Improving the accuracy of intrusion detection using gar-forest with feature selection," in *Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015*. Springer, 2016, pp. 539–547.

[87] P. Krömer, J. Platoš, V. Snášel, and A. Abraham, "Fuzzy classification by evolutionary algorithms," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2011, pp. 313–318.

[88] H. Benaddi, K. Ibrahimi, and A. Benslimane, "Improving the intrusion detection system for nsl-kdd dataset based on pca-fuzzy clustering-knn," in *2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)*. IEEE, 2018, pp. 1–6.

[89] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, and C. D. Perkasa, "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert systems with Applications*, vol. 38, no. 1, pp. 306–313, 2011.

[90] C. Guo, Y. Ping, N. Liu, and S.-S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, 2016.

[91] P. Bedi, N. Gupta, and V. Jindal, "Siam-ids: Handling class imbalance problem in intrusion detection systems using siamese neural network," *Procedia Computer Science*, vol. 171, pp. 780–789, 2020.

TREEPOP WISANWANICHTHAN received the BS degree in Computer Science and Mathematics from the University of New South Wales, Australia, in 2019. He is currently a Cyber Security officer at the Royal Thai Air Force. His research interest is on applied machine learning in various domains such as cyber security.

MASON THAMMAWICHAI received the BS degree in Computer Engineering from University of Wisconsin-Madison, USA and the MSc in Avionic System from University of Sheffield, UK and a PhD from Imperial College London in 2016. He has been a faculty member at the graduate school of Navaminda Kasatriyadhiraj Royal Air Force Academy since 2016. From 2017 to 2021, he was an Assistant Professor. Since 2021, he has been an Associate Professor. His research interest is on optimization, optimal control, UAV, swarm robots, intelligence systems, machine learning, deep learning and cyber security.