A two-layered collaborative approach for network intrusion detection system using blended shallow learning gaussian naïve bayes and support vector machine models

Nilesh Ghanshyam Pardeshi [a,1,*], Dipak Vitthalrao Patil [b,2]

[a] MET's Institute of Engineering, Nashik, Savitribai Phule Pune University, Pune, India
[b] GES R. H. Sapat College of Engineering, Management Studies and Research, Nashik, Savitribai Phule Pune University, Pune, India
[1] ngpardeshi@gmail.com; [2] dipakvpatil17@gmail.com
* corresponding author

ARTICLE INFO

Article history
Received March 18, 2025
Revised June 14, 2025
Accepted June 27, 2025
Available online August 31, 2025

Keywords
Two-layered collaborative approach
Cross-sectional correlation
Gaussian naïve bayes
Intrusion detection system
Support vector machine

ABSTRACT

The majority of network intrusion detection systems use a signature matching technique. To detect abnormalities and unfamiliar attacks using machine learning methods is a more reliable approach. However, due to significant variations in attack trends, applying a single classifier is impractical for the effective detection of all types and forms of attacks, particularly rare attacks such as User2Root (U2R) and Remote2Local (R2L). Consequently, a hybrid strategy is expected to provide more promising performance. The proposed Two-Layered Collaborative Approach (TLCA) particularly addresses the problem as mentioned earlier. Principal Component Analysis optimizes variables to handle the variation resulting from every kind of attack. The proposed method investigates several types of attacks and discovered that the behaviors of U2R and R2L attacks are similar to those of regular users' characteristics. To identify DoS and Probe attacks, TLCA uses a Shallow Learning classifier, such as Gaussian Naïve Bayes, as Layer 1. Similarly, the Support Vector Machine at Layer 2 discriminates between U2R and R2L and typical occurrences. We have divided KDDTrain+ into Set 1 and Set 2 by observing that it involves two 2-dimensional PCA analyses. Cross-sectional Correlated Feature Selection (CCFS) is employed to choose key attributes. PCA and KPCA are applied to datasets to reduce dimensionality. The results obtained using the proposed method on the NSL-KDD dataset are compared with those of available benchmark methods. According to the experimental data, TLCA outperforms all single machine learning classifiers and surpasses many current cutting-edge IDS approaches. The proposed method achieves detection rates of 92.4%, 92.3%, 95.6%, and 100% for DoS, Probe, R2L, and U2R, respectively. The proposed TLCA also demonstrates a better ability to identify unusual attacks. It also yields improved detection rate results for known attacks, at 94.1%, and for unknown attacks, at 91.1%, when using the KDDTest+ dataset for testing.

1. Introduction

A significant increase in attacks on computers and network-based services over the past couple decades has made cyber security an important topic for safeguarding systems against threats both locally and globally [1]. Even though network firewalls and data encryption have already met the requirements for fundamental security and provided basic security for computers and networks, many threats remain unreported, thereby affecting the services overall [2], [3]. Because incursion threats are hazardous, they

need to be addressed immediately. Organizations are most vulnerable to attackers, particularly those that require a high level of protection, such as airports and military bases. Inability to find an invader automatically results in security breaches such as the loss of confidential data, gaining illegal access, and posing as an administrator with malevolent intent. There are four main attack classes, according to NSL-KDD [4]. 1) A denial of service (DoS) assault bombards the intended recipient with a tremendous volume of traffic to temporarily disable the service. 2) A probe attack is one that searches for and utilizes network flaws in accessible ports to discover amenities used by the victim. 3) Remote 2 Local (R2L) attack aims to take advantage of the victim's weaknesses to obtain unauthorized entry to nearby networks. 4) User 2 Root (U2R) attack seeks to use a machine's flaws to get root access or seize command of the machine. U2R and R2L attacks are rare but have a more negative impact on a system [5].

To create an anomaly-based IDS, different kinds of machine learning algorithms have been investigated and put into practice [6]–[8]. Two Machine Learning approaches are frequently used in the field of IDS. 1) Supervised learning, which builds a function of mapping from input-output pairs that have been pre-defined; and 2) Unsupervised machine learning, enables a framework to find inner linkages on its own. The most popular method in IDS is supervised machine learning. Decision Tree, Support Vector Machine, K-Nearest Neighbors, and Naive Bayes are a few examples [9]–[12]. Unsupervised machine learning is most often used to describe clustering techniques, such as K-Means [13].

The selection of pertinent attributes in the event of many attack types presents the primary issue in developing an effective IDS. Consequently, to eliminate noise and uninformative features, feature selection is an essential process. Feature selection is a crucial component for increasing IDS accuracy [14]. To enhance the outcomes of classification, many IDS researchers are investigating the greatest attribute extraction techniques, like utilizing the Local Search technique using K-Means algorithm, Particle Swarm Optimization, Genetic Algorithm, Correlation Coefficient, and Ant Colony Algorithm [15], [16]. Deep learning and Artificial neural networks have been effectively employed recently to cope with intricate designs, specifically in the handling of languages and images. Convolutional neural networks, as well as recurrent neural networks, have been used in studies to solve IDS problems [17]–[19]. Each Machine Learning algorithm has unique abilities. While some algorithms are adept at seeing a certain kind of attack, others are not [20], [21]. Recently, methods that incorporate two or more learning algorithms have been suggested because they perform better at identifying different types of attacks [21]. Popular learning algorithms for IDS, like the ensemble technique, typically produce superior results to single estimators. Multiple base classifiers are integrated using an ensemble learning technique, such as Random Forest (RF), to improve prediction performance [22].

While individual components such as layered classification (using GNB and SVM), correlation-based feature selection, and dimensionality reduction via PCA/KPCA have been explored in prior IDS literature, our work introduces a uniquely orchestrated combination of these techniques. Unlike existing hybrid or ensemble models, our system, TLCA, integrates these components in a carefully designed multi-layer structure, where each layer is trained on behaviorally distinct attack groups and optimized with context-aware feature selection (i.e., CCFS), followed by appropriate dimensionality reduction. This tailored configuration improves detection performance, particularly for low-frequency but high-impact attacks (R2L and U2R) while maintaining generalizability. Thus, the contribution of our approach lies not only in performance gains but also in the strategic interplay of components, which, to the best of our knowledge, has not been demonstrated in previous IDS frameworks.

To address the previously outlined challenges, the proposed system extends the work done by Wisanwanichthan *et al.* [1]: (I) In contrast to the ensemble method and a single ML classifier, we presented a two-layered collaborative approach (TLCA). The suggested method consists of two levels that operate in a cascade fashion; layer 1 is used to identify Probe and DoS using a shallow learning model, such as Gaussian Naïve Bayes at layer 1, while layer 2 is used to find R2L, U2R, and Normal using an SVM model. (II) Using PCA, for conducting data analysis, we discovered that R2L and U2R behave in a manner that is consistent with typical traffic patterns, whereas DoS and Probe differ considerably from the others. This research motivated us to create TLCA. (III) The NSL-KDD training

dataset was split into two sets: Set 1, which contains all classes, and Set 2, which contains only R2L, U2R, and Normal classes. This division of the data set into two sets makes the approach unique. These were used to independently train the two classifiers so that one could be used to recognize unusual attacks, such as U2R and R2L, in addition to regular connections. (IV) We used correlation coefficients to present Cross-sectional Correlated Feature Selection (CCFS). (V) The proposed system uses Kernel PCA at layer 2 for dimensionality reduction of nonlinear data in Set 2 and obtained improved results than stated in [1]. (VI) We evaluated our suggested strategy to demonstrate that, when measured against numerous other cutting-edge methods now in use, TLCA offers superior detection rates and false alarm rates on the results of low-frequency attacks as well as overall effectiveness. Also, detection rates of TLCA on both recognized and unseen attack categories are improved in comparison with existing works. (VII) We demonstrated that TLCA is a highly competitive hybrid method and that it executes noticeably better than the usual individual ML strategies. While TLCA incorporates known techniques, its novelty lies in the coordinated layering of GNB and SVM with class-specific feature selection via CCFS. This structured interaction enables each layer to specialize in distinct attack behaviors, leading to improved detection of both frequent and rare threats.

Wisanwanichthan *et al.* [1] proposed DLHA, which uses a hybrid approach by creating two different models for recognizing attacks. The first model recognizes DoS and Probe attacks, while another model recognizes R2L, U2R, and Normal. A standard dataset for intrusion detection study, NSL-KDD, is used by Sukhadeo *et al.* [23] to train and evaluate the suggested model. To assess how well these classifiers performed, the data were divided into four feature groups derived from the NSL-KDD dataset. The recommended approach provides better detection rates and fewer false positives than conventional rule-based frameworks. Immanuel *et al.* [24] train and assess their framework using the suggested stacked ensemble machine learning framework, comparing it against traditional machine learning techniques and previous studies, which utilize different performance parameters, on the NSL-KDD dataset. Ali *et al.* [25] have provided a summary of IDS in their research work, along with information on its classes, techniques, detected attacks, datasets, and performance measures. Qazi *et al.* [26], have built a deep learning-powered blended intrusion detection framework that identifies network threats using a convolutional recurrent neural network. In the proposed Blended Deep-Learning-powered Network Intrusion Detection Framework, a deep-layered recurrent neural network selects the features, while a convolutional neural network performs a convolution to gather local characteristics. Almutairi *et al.* [27] employed several machine learning models, including Random Forest, J48, Support Vector Machine, and Naïve Bayes, with binary and multi-class categorization, to assess Network Intrusion Detection Systems using the NSL-KDD standard dataset.

These days, many anomaly-based intrusion detection frameworks utilize hybrid machine learning models, as they enhance efficiency and performance. A hybrid feature selection approach for a multi-level data mining model was suggested by Yao *et al.* [21]. The researchers conducted several tests to determine which Machine Learning algorithms would be most effective in identifying each type of attack. The ultimate approach of detection included four distinct classification algorithms: a linear SVM for DoS detection, an ANN with a logistic activation function for Probe detection, an ANN with a ReLU activation function for R2L detection, and an ANN with an identity activation function for U2R detection. On the other hand, the usage of various classifiers for various data sources led to longer computing times throughout both the training and testing operations [28]. In [29], GA-SVM, which combines SVM and Genetic Algorithm, was introduced. Based on three objectives, the genetic algorithm was utilized as a feature minimization strategy to minimize the number of attributes from 45 to 10. The GA utilized variation and crossover to generate the optimal feature subcategories for SVM training. A model of Deep Hierarchical Network with combined sampling was proposed by Jiang *et al.* [17]. Single-Side Decision Making was employed to reduce the sample size in the most common class categories, while the Synthetic Minority Oversampling method was utilized to increase the sample size in the underrepresented groups, aiming to balance the class distribution.

A Double-layered Dimension reduction and dual-tier classification model was suggested by Pajouh *et al.* [30] to detect harmful behaviors, specifically U2R and R2L. The certainty factor of the KNN

algorithm and the Naive Bayes algorithm were used in the dual-tier categorizing system. Tama *et al.* [14] provided a Dual-Stage Ensemble technique which used the Ant Colony Algorithm, Genetic Algorithm and Particle Swarm Optimization feature choice algorithms. To choose the basis classifiers, the authors tested several classifiers, such as Deep Neural Networks, Decision Tree, Random Forest, and KNN [31]. A composite architecture utilizing Probabilistic Self Organizing Maps, Fisher Discriminant Ratio and PCA was suggested by Hoz *et al.* [32]. Anomaly-containing cases were found using the PSOMs technique. In [33], the Auto-Encoder intelligent IDS was suggested. To perform attribute selection, the authors eliminated attributes that include zeros that are greater than 80%. IDS based on recurrent neural networks was presented in [20]. By changing the learning rate and the number of hidden nodes, the authors applied optimal parameters and one-hot encoding. Multi-level hybrid data mining is a method proposed by Gogoi *et al.* [34]. It has three stages, with the first stage using supervised machine learning CatSub+ to categorize Probe and DoS, the second stage using unsupervised machine learning K-point method to identify usual traffic, and the third stage using anomaly-based classifier GBBK to categorize U2R and R2L

The primary differentiation between hybrid methods and earlier comparable research works lies in the selection of features. While many strategies select features depending on the attributes that are most appropriate for every attack, executing attribute selection on a particular attack type is more suitable. A hybrid design is still another important differentiation. Four classifiers were used by the authors in [21], to find various types of assaults, which enhanced functionality but slowed down the procedure. The two-tier hybrid IDS described by Pervez and Farid [35] uses two classifiers with the best attributes extracted from LDA and PCA, in contrast. Still, the dual-tier IDS experienced an R2L recognizing functionality problem. As a result, previous articles did not advance the use of an efficient choice of features or a more effective hybrid IDS architecture [36]–[38].

While several hybrid and layered IDS models have been proposed, many struggle with effectively detecting low-frequency, behaviorally complex attacks such as R2L and U2R [39]–[41]. Despite advancements in feature selection and ensemble techniques, existing approaches often fail to align classifier choice with the nature of specific attack types, resulting in performance degradation on rare classes. In contrast, the proposed TLCA framework introduces a structured, two-layer design that explicitly maps classifier capabilities to the behavioral traits of attack groups. Layer 1 utilizes GNB to efficiently model high-volume, linearly separable attacks, such as DoS and Probe. At the same time, Layer 2 employs an SVM with kernel-based dimensionality reduction to capture the nonlinear, sparse patterns associated with R2L and U2R instances. Central to this design is the CCFS strategy, which extracts layer-specific features based on class groupings, ensuring that each classifier operates within a tailored feature space. This targeted, feature-driven layering distinguishes TLCA from previous IDS models that apply generalized classifiers or feature sets across all classes, offering a more adaptive and effective defense against a diverse range of network intrusions.

The various methods applied by different researchers on Intrusion Detection Systems are shown in Table 1. The table presents various methods applied by different researchers on Intrusion Detection Systems. The presented methods are a hybridization of different techniques for IDS. The table also presents a comparison of techniques based on feature selection method, performance measures, and a summary of the method applied. To clearly define the scope of this study, we aim to address the following research questions:

- Can a two-layered IDS architecture improve detection for both frequent (DoS/Probe) and rare (R2L/U2R) attacks?
- Does the proposed Cross-sectional Correlated Feature Selection (CCFS) enhance classifier performance?
- How does the TLCA compare to existing hybrid and single-classifier NIDS on the NSL-KDD dataset?

Table 1. Comparison between this research and the previous researches

Authors	Year	Suggested Method	Feature Choice	ML&DL Algorithms	Performance Parameters	Summary of the method applied
Liu *et al.* [39]	2021	Combined Kmeans+RF	Attribute Ratio	RF, KMeans, CNN+LSTM	Accuracy, TPR	Adaptable Random Forest, K-Means, and CNN with LSTM are used in the proposed hybrid IDS.
Wisanwanichthan *et al.* [1]	2021	DLHA	ICFS and PCA	SVM and NBC	Detection Rate, FAR, Accuracy, Precision, F1 Score	A dual-layered hybrid IDS using SVM and NBC is proposed. PCA and ICFS are used for feature choice
Immanuel *et al.* [24]	2022	Stacked Ensemble	Random Forest	RF, KNN	Accuracy, Precision, F1 Score, Recall	trained and evaluated their model using the proposed stacked ensembled machine learning model, and with the NSL-KDD dataset
Manjunath *et al.* [2]	2023	CRNN	NA	CNN, RNN	Accuracy	Ensemble learning methods in machine learning algorithms are used to detect and prevent malicious packets in the network
Qazi *et al.* [26]	2023	HDLNIDS	CNN	CNN, RNN	Accuracy, Precision, F1 Score, Recall	The CNN performs the convolution to collect local features, while a deep-layered RNN extracts the features in the proposed Hybrid Deep-Learning-Based Network Intrusion Detection System.
Kasongo et al [42]	2023	RNN	XGBoost	LSTM, GRU, Simple RNN	Validation and Test Accuracy, F1 Score	presents an IDS framework based on various Recurrent Neural Network architectures, including LSTM, GRU, and Simple RNN
Ben *et al.* [43]	2023	Hybrid Model	NA	CNN and BiLSTM	Accuracy, Precision, F1 Score, Recall	proposes a hybrid model combining Convolutional Neural Networks with Bidirectional LSTM to improve network intrusion detection
Hnamte *et al.* [44]	2023	Hybrid DL based NIDS	NA	DNN, CNN, LSTM	Accuracy, Training Time	presents a deep learning-based Network Intrusion Detection System designed for intelligent and efficient attack detection
Sajid *et al.* [45]	2024	Hybrid Model	XGBOOST, CNN	LSTM	Accuracy, Training Time, Precision, F1 Score	The model employs XGBoost and CNNs for feature extraction, integrated with LSTM networks for classification
Zhengfa Li *et al.* [46]	2024	VAE-WGAN model	Stacked LSTM and MSCNN	LSTM, CNN	Accuracy, Precision, F1 Score, Recall	proposes the VAE-WGAN model for generating labeled synthetic data to facilitate class balancing. For intrusion detection, a hybrid model integrating stacked LSTM and Multi-Scale CNN is employed.
Rajathi *et al.* [47]	2025	HLM	NA	KNN. DT, RFGBM, SVM, LR, NBC, LDA	accuracy, precision, recall, F1-Score, ROC, Detection Rate, FAR.	Proposes a Hybrid Learning Model that integrates non-parametric and parametric classifiers
N G Pardeshi *et al.* (This Paper)	2025	TLCA	CCFS, PCA, KPCA	Gaussian Naïve Bayes and SVM	Accuracy, FAR, Precision, F1 Score, Detection Rate	Proposed Two-layered Collaborative IDS using Gaussian Naïve Bayes and SVM.

2. Method

2.1. Dataset Description

KDD99, which collected TCP dump information from the DARPA98 program for off-line intrusion detection assessment, was the most extensively utilized dataset in assessing anomaly-based intrusion detection systems. But there are several built-in issues with the KDD99. As a result, this work utilizes the NSL-KDD dataset instead. To address the skewed and unevenly distributed KDD99 data collection, the NSL-KDD was proposed in 2009 [4]. Because so many redundant and duplicate pieces of data are eliminated, the NSL-KDD has many advantages over the obsolete KDD99. Additionally, well-chosen cases are well-represented; that is, the distribution of attacks of different levels of complexity in the training as well as testing sets is uniform, and the quantities of attacks and regular instances are not that different from one another. Comparing anomaly-based algorithms utilizing various ML techniques as a result yields more trustworthy classification findings [4], [36]. As illustrated in Table 2, attacks fall into four major groups, as per NSL-KDD.

The NSL-KDD is split up into five classes: Probe, DoS, U2R, R2L, and Normal. Despite being an improved form of KDD99, the NSL-KDD still exhibits an inherited unequal distribution of classes in the datasets. For instance, it has been found that normal records account for most cases in the training data set (53.46%), followed by DoS share, which is 36.46%, and Probe share, which is 9.25%, while R2L and U2R shares are 0.79% and 0.04% respectively, which are extremely rare. Due to the model's bias, the issue is that if only one framework is used, it will be unable to efficiently identify U2R and R2L [36]. In comparison to Probe and DoS attacks, U2R and R2L attacks are more dangerous [5].

2.2. Analysis of Class Diversity

There are 41 attributes per instance in the NSL-KDD. The attributes can be categorized into the following four groups: Intrinsic features (attributes 1-9) extracted from the packet header, Content features (attributes 10-22) include the actual packet payloads, time-based features (attributes 23-31) are derived from records of traffic connection intervals of 2 seconds, and characteristics depending on hosts (attributes 32–41) resemble features that are dependent on time, but instead consider all series of connections. These characteristics help to evaluate attacks that last more than two seconds. Three features, flag, protocol_type, and service, are categorical, and 39 of the features are numerical.

To execute data examination using training sets, initially, we implemented pre-processing of the data by allocating numerical label tags to [0, 1, 2, 3, 4] instead of [Normal, DoS, Probe, R2L, U2R] correspondingly. We then use those categorical attributes to apply one-hot encoding. One-hot encoding is an efficient method for encoding categorical details after transforming a categorical characteristic into a numeric value. It does, however, presuppose that each value in the category has zero associations. There is no internal hierarchy or link among the category features. New features are produced by a one-hot encoder with a vector binary representation for each unique original value, where n is the number of distinct values in a feature. For example, the encoding for the TCP protocol is [0,1,0], whereas the ICMP protocol is [1,0,0], where a value's presence is denoted by a 1 and its absence by a 0. After preprocessing the data, we had a total of 122 attributes, but the num_outbound_cmds attribute had zero values, indicating that it lacked the ability to forecast. Consequently, the attribute was removed. We limited our consideration to 121 attributes in this study. On the training data, we conducted a study of the two-dimensional PCA scatter plot, as shown in Fig 1.

In Fig. 1, we assigned the following labels: DoS to green, Probe to blue, R2L to light blue, U2R to yellow, and Normal to red. We did not include Normal in the top graph. Naturally, most of the DoS and Probe instances are dispersed from typical instances, but the majority of U2R and R2L attacks cross over with one another as well as the typical connections. Stated differently, they acted more comparably to one another than those distant attacks, such as Probe and DoS, which suggests that U2R and R2L invaders shared a few traits. According to the graph at the bottom, the bulk of Probe and DoS attacks are mostly autonomous of the other attacks, with only a little amount of overlap in the top region. Additionally, a small number of Probe and DoS instances cross over with regular links. It is obvious why

many IDS techniques failed to detect U2R and R2L threats accurately, which further resulted in a significant false alarm rate due to their behavioral resemblance to typical network connections. The data we collected based on the PCA interpretation and other research proves that their algorithms do an excellent work of finding DoS and Probe but fight with poor detection rates on under-represented attacks. suggesting that a rigorous detection method is necessary for U2R and R2L attacks. To solve this specific issue, we created TLCA.

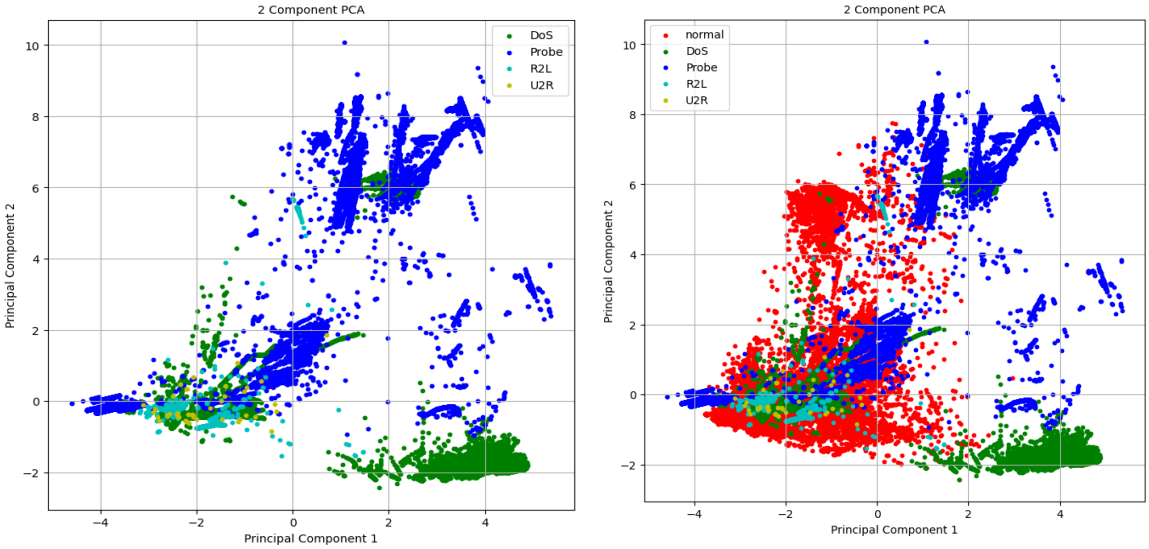

Fig. 1. Presents distribution of classes in KDD Training dataset. The classes are Normal, Dos, Probe, R2L, and U2R. Two-dimensional PCA scatter plot study was conducted on the training dataset and 2D-PCA illustration of KDDTrain+ divided according to classes is presented here

Table 2. Principal attack types in the NSL-KDD dataset

Types of Attacks	Attack Definition
Denial of Service (DoS)	To overload connections so that the service is unavailable
Probe	To obtain crucial information (port scanning)
Remote to Local (R2L)	To allow remote machines to access local resources
User to Root (U2R)	To obtain Super User rights

2.3. Proposed Framework of TLCA

Here, we described the framework overview of our proposed approach, shown in Fig. 2. Data preparation, data translation, and training processes are the three key steps. Then, we showed how the TLCA abnormality-based intrusion detection system operates to quickly identify suspicious network connections. Our strategy is particularly distinctive in that we were the first to use Cross-sectional Correlated Feature Selection (CCFS), which selects crossing over attributes from various attacks made in opposition to others. Our framework also features two levels of detection, with Level 1 used to distinguish between Probe and DoS attacks and other types of attacks across all connections. We then have a dedicated classifier at Level 2 that focuses on recognizing U2R and R2L attacks.

2.3.1. Theoretical Framework of TLCA

According to the previous research, R2L and U2R assaults were more akin to typical connections, while the maximum of DoS and Probe attacks greatly varied from the typical patterns. For a real-time IDS, we created a conceptual model that specified that it should include two classifiers. The initial classifier must be precise and quick to handle several network connections at once. The Gaussian Naive Bayes Classification algorithm is chosen due to its effectiveness and dependability [12]. The Support Vector Machine classification algorithm is utilized at level 2. To tackle non-linearly separable problems, it provides a Radial Basis Function kernel, which is a useful tool for observing the difference between R2L, U2R, and typical cases.

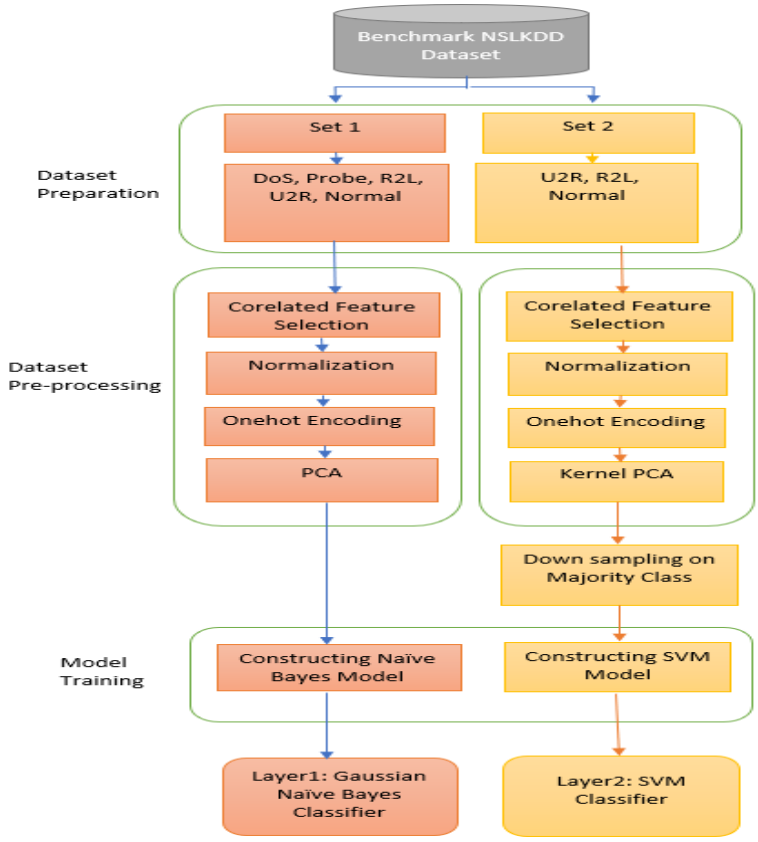

Fig. 2. An abstract structure of TLCA anomaly-based intrusion detection system

2.3.2. Preparation and Transformation of Data

The proposed system consists of two layers, each with distinct capabilities. During the data preparation phase, two sets of data are made by using the initial NSL-KDD training dataset. All records and categories are included in the first set, whereas Normal, R2L, and U2R records are the only ones in the second set. The second step involves implementing CCFS, Min-Max normalization, one-hot encoding, PCA, and Kernel PCA. The process of selecting a subset of relevant attributes and excluding irrelevant features is known as feature selection. In addition to enhancing accuracy, it considerably increases processing speed. Still, in cases where there are a greater number of categories in the dataset, it could be hard to pick the accurate characteristics because a few attributes that are effective for one sort of attack may not be predictive for others. Additionally, it has been exhibited that various attributes impact various attacks since attack trends differ [3], [37]. For instance, a DoS attack is probably going to use the TCP protocol. Selecting unwanted attributes always makes IDS unproductive. We provided CCFS to sort out this issue, shown in Fig. 3.

During feature selection, we employed the Pearson Correlation Coefficient (PCC) to identify the most relevant attributes for each dataset subset. PCC is a bivariate statistical measure that assesses the linear correlation between two numerical variables, offering low computational complexity and the ability to handle high-dimensional data. It is computed by dividing the covariance of the variables by the product of their standard deviations. To tailor the feature selection process, we applied a PCC threshold of 0.1 for Set 1 and 0.01 for Set 2—values empirically determined based on validation performance. Set 1, which includes all classes, uses a broader threshold due to the greater separability of DoS and Probe attacks. In contrast, Set 2 (Normal, R2L, U2R) required a more stringent threshold to better differentiate among overlapping behaviors. Importantly, features were selected using a union-based approach, capturing attributes correlated with any attack type within each set, rather than limiting selection to their intersection.

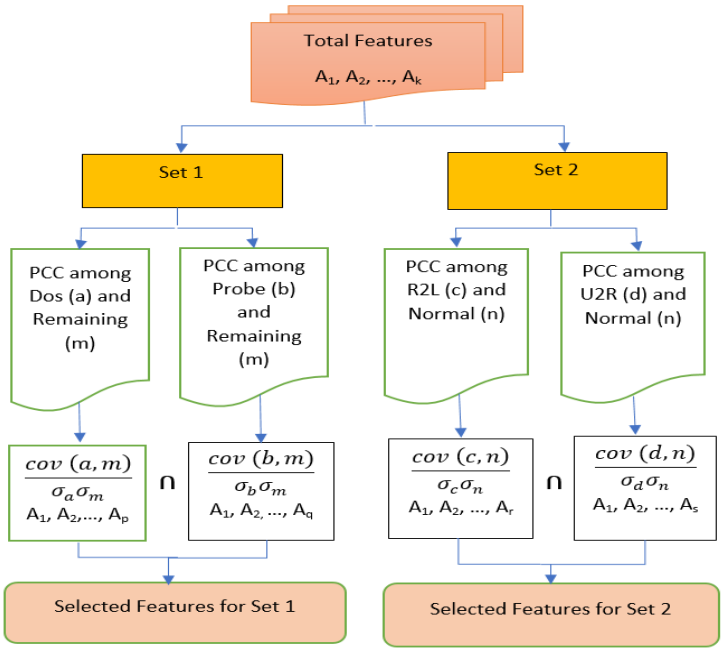

Fig. 3. Cross-sectional Correlated Feature Selection (CCFS)

PCC can be expressed as follows where $X = [x_1, x_2, \dots, x_n]$ and $Y = [y_1, y_2, \dots, y_n]$ are arbitrary vectors with n number of samples.

$$\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y} \tag{1}$$

therefore, it can be calculated as

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

whereas n is the number of instances, Standard deviation $\sigma_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$, Mean $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$, and Covariance between x and y $cov(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$

Let A be the attributes from the training set $\{A_1, A_2, \dots, A_k\}$. In Set 1, we gave DoS and Probe a score of 1, and the others a score of 0. Let A(DOS) = $\{A_1, A_2, \dots, A_p\}$ be the attributes that distinguish DoS from the other characteristics and have a PCC greater than 0.1. Let the attributes with PCC larger than 0.1 between Probe and the other be A(Probe) = $\{A_1, A_2, \dots, A_q\}$. A(DOS) and A(Probe) are predictive attributes that classify DoS from the other and Probe from the other, respectively. To distinguish DoS and Probe from the others, A(DOS) ⋂ A(Probe) are frequent predictive attributes. As a result, set 1's chosen characteristics are A(DOS) and A(Probe). Since most attributes are uncorrelated, we applied the same for Set 2 but with a 0.01 threshold.

R2L and U2R were assigned a 1 in Set 2, while regular records were assigned a 0. Next, PCC was determined between Normal and R2L as well as between Normal and U2R. Therefore, the chosen features for Set 2 are F(R2L) ⋂ F(U2R). The primary goal of CCFS is to exclude from the groupings any noticeable uncorrelated features. Since the data's standard deviations were quite low after the CCFS, we normalized them to fall within the span [0,100]. Min-Max Normalization can be done using a formula in (3).

$$x'_{ij} = \frac{x_{ij} - \min(x)_j}{\max(x)_j - \min(x)_j}, i = 1, 2, \dots m; j = 1, 2, \dots, n \tag{3}$$

We then carried out one-hot encoding, PCA and Kernel PCA in that order. PCA was applied to reduce the high-dimensional Set 1 and Set 2 data to lower dimensional, uncorrelated, linearly transformed data that had significant variation. As Set 2 contain mostly nonlinear data containing R2l, U2R and Normal sharing more complex boundaries, Kernel PCA with RBF kernel is applied to Set 2 Dataset only for reducing its dimensionality.

Kernel-PCA is a PCA extension that employs kernels to enable the separability of nonlinear data, and further reduce nonlinear dimensionality. Its fundamental premise is to map the linearly inseparable data onto a space in higher dimensions where it is made separable linearly. KPCA is a non-linear PCA approach based on kernel functions that performs linear PCA in F after creating a non-linear mapping $\emptyset(x)$ from the input space x to the feature space F by means of a non-linear conversion $\emptyset$. This enables it to detect more complex and non-linear correlations between the data points. Since the radial basis function performs well, it is frequently selected as the kernel function of KPCA in practical applications.

The covariance matrix of projected data, C:

$$C = \frac{1}{N}\sum_{i=1}^{N} \emptyset(x_i)\emptyset(x_i)^T \quad (4)$$

The eigen vectors and eigen values of are given by:

$$C_{vk} = \lambda_k v_k \quad (5)$$

Where k = 1, 2, ..., N

Among two input samples, x and y, in the original space, it is possible to avoid carrying out the nonlinear mappings and calculating both the dot products in the feature space by utilizing a kernel function like.

$$k(x, y) = \phi(x).\phi(y)$$

Kernel PCA can be summarized as a 4-step process:

- Develop the kernel matrix K from the training dataset

  $K[i][j] = k(x_i, x_j)$

- If the estimated dataset $\phi(x_i)$ possesses no zero mean then use the gram matrix $\overline{K}$ to replace the kernel matrix K.

  $\overline{K} = K - 1_N K - K1_N + 1_N K 1_N$

  Where as 1_N is a $N \times N$ matrix with all entries equivalent to 1_N

- Obtained the eigen vectors a_k of the kernel matrix K

  $K_{ak} = \lambda_k N_{ak}$

- Compute kernel principal component $yk(x)$

  $$y_k(x) = \phi(x)^T v_k = \sum_{i=1}^{N} a_{ki} k(x_i, x_j) \quad (6)$$

We employ the minimum number of features necessary to create an effective IDS. As a result, we choose the lowest value that can preserve 95% of the variance. For each group, since the cases were distinct, a separate data transformation was conducted. As a result, the scaling coefficients, principal component count, and features that were chosen varied. There are two different types of data transforms. One-hot encoding is an efficient method for protecting anticipative details after transforming a categorical characteristic to a numeric value. Data balance in the training dataset after the data transformation is essential to prevent bias towards many records. Notably, the ratio of U2R+ R2L =

1,047 cases to Normal = 67,343 cases is almost 64:1. The majority class must be down sampled to avoid bias. To make the ratio 1:1, for instance, 1,047 normal examples were chosen at random. The down-sampling procedure was not required because Set 1's class ratio is not excessive.

2.3.3. Model Training

The training step is vital. A shallow learning algorithm, such as Gaussian Naive Bayes, is chosen for the classification of Set 1. A Support Vector Machine is selected for the categorization of Set 2.

- Gaussian Naive Bayes Algorithm

  Gaussian Naive Bayes Algorithm is a straightforward yet effective probabilistic approximation that works by using the Bayes theorem under the presumption that all the attributes being considered are separate from one another. i.e., that each factor impacts the outcome in a different way. The role of the NBC in our suggested technique is to find Probe and DoS. To achieve this, Probe and DoS attacks are designated as 1, while the others are designated as 0. Assume that the dependent feature vector within the information is x, such that $x = \{x_i, x_2, \dots, x_n\}$, and that $y = \{y_i, y_2\}$ = {Rest, DoS/Probe}. The Bayes' theorem can be expressed according to equation 7:

$$P(y = k|x_1, \dots x_n) = \frac{P(y=k)P(x_1,\dots,x_n|y=k)}{P(x_1,\dots,x_n)} \quad (7)$$

  P(y) represents the earlier likelihood, $P\ (x_i, x_2, \dots, x_n|y)$ is a reliable vector's likelihood in relation to its category, and P $(x_i, x_2, \dots, x_n)$ is the smallest probability or evidence of a reliable vector. The probability in the past of y occurring, provided that $(x_i, x_2, \dots, x_n)$ has taken place is $P(y\ |x_i, x_2, \dots, x_n)$. It can be described as follows under the conditional premise that each attribute is independent of the others.

$$P(y = k|x_1, \dots x_n) = \frac{P(y=k)\prod_{i=1}^{n} P(x_i|y=k)}{P(x_1)*P(x_2)\dots*P(x_n)} \quad (8)$$

  where n denotes the quantity of attributes remaining after data preprocessing on Set 1. $P\ (x_i, x_2, \dots, x_n)$ being a consistent across all. Therefore, the Naive Bayes Classifier has the following categorization expression.

$$y' = argmaxP(y = k)\prod_{i=1}^{n} P(x_i|y = k) \quad (9)$$

  As the NBC uses the Gaussian technique for classification, the $P\ (xi \mid y = k)$ is taken to be Gaussian in the manner described below:

$$P(x_i|y) = \frac{1}{\sqrt{2\prod\sigma_y^2}}\exp\left(\frac{-(x_i-\mu_y)^2}{2\sigma_y^2}\right) \quad (10)$$

  The NBC has shown amazing categorization outcomes in the IDS issue, even though in real-world applications, the feature-wise independence assumption is violated virtually often

- Support Vector Machine algorithm with Linear and RBF Kernel

  The very often utilized supervised Machine Learning algorithm for categorization applications is SVM. To distinguish two classes with the greatest possible margin, SVM finds the optimum hyperplane in a large-dimensional area. By permitting selections of kernels, such as linear and radial basis function, it offers flexibility in implementations. RBF, which is a kernel of nonlinear support vector machine algorithm, excels at classifying U2R and R2L from normal network connections because it deals with data that share complex boundaries. SVM hyperparameters were optimized using stratified 5-fold cross-validation. We tuned the regularization parameter (C) and kernel coefficient (γ) over a logarithmic grid: $C \in \{0.1, 1, 10, 100\}$, $\gamma \in \{0.001, 0.01, 0.1, 1\}$. The best configuration was selected based on maximum average F1-score across folds.

Regarding any specific connection-class pair of training vectors $(xi, yi), i = 1, 2, \ldots, n$ wherein $xi \epsilon Rn \; and \; y \epsilon \{1, 0\}^n$, where a positive category is represented by 1 and a negative category by 0. The following issue needs to be resolved for SVM to work:

$$min \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi i \tag{11}$$

Subject to $y_i(w.x_i + b) \geqq 1 - \xi_i$

$$\xi_i \geqq 0, i = 1, 2, \ldots, n \tag{12}$$

The aim of the formulation is to optimize the difference between the two categories by reducing $||w||^2$. Strength of the penalty C is used to control samples that are incorrectly classified at an interval ξ_i from the proper margin border that fits to the value $y_i(w.x_i + b) \geqq 1 - \xi_i$. For every sample x, the decision function output is understood to be:

$$\sum_{i \epsilon SV} y_i \alpha_i K(x_i, x) + b \tag{13}$$

The analogous class from the prediction is its sign. In this work, the SVC kernels selected for validation are RBF and linear. If an RBF kernel that is non-linear could consistently outperform its linear version, it has never been proven. To observe the U2R and R2L border, we used RBF and linear as the two kernels for the adjustment of parameters. We only adjusted the SVM's hyperparameters in the training dataset, i.e., KDDTrain+, with the help of 10-fold stratified cross-validation, so that data breach and overfitting of the dataset can be avoided. The variables in question are C and gamma. C, a regularization parameter that increases the penalty with each sample that is incorrectly categorized, is provided for both linear and RBF. A single training sample's radius of influence is governed by the RBF gamma. The parameters are: C = 0.01, 0.1, 1, 10, 100, 1000 for the linear case, C = 0.1, 1, 10, 100 for the RBF case, and gamma = 0.01, 0.1, 1, 10. As a result, the linear kernel has six specifications, and the RBF kernel has 16 specification sets. The computer's specs include 8GB of RAM, Processor Intel(R) Core (TM) i5-10210U CPU @ 1.60GHz, 2112 MHz, 4 Core(s), 8 Logical Processor(s), and Ubuntu 20.04 LTS.

2.3.4. TLCA Algorithm

Fig. 4 illustrates real-time traffic categorization using TLCA. TLCA is suggested to enhance the overall rate of detection, and particularly the rate of detection of uncommon attacks, which are more aggressive, such as U2R and R2L in this research.

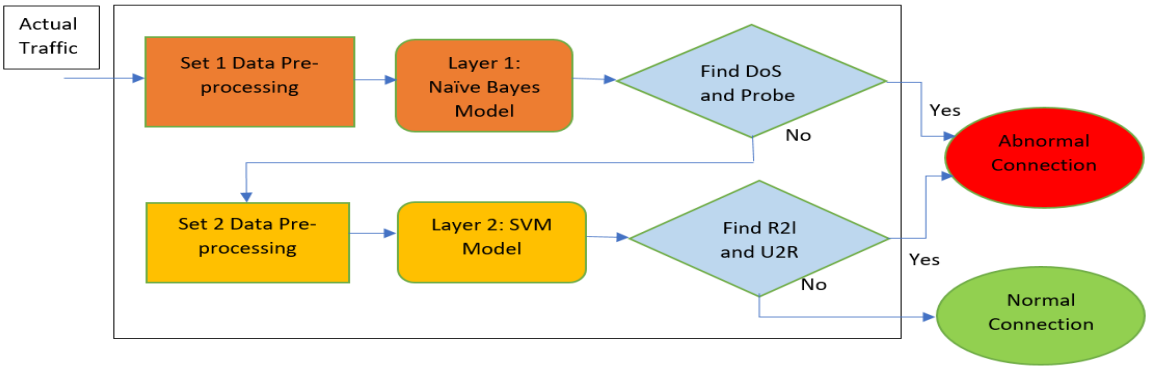

Fig. 4. Classification of traffic in real time using TLCA

Since the proposed system has CCFS, PCA, and KPCA to minimize data dimensionality, it is also intended to be an effective IDS in real-time. The way the TLCA algorithm operates is as shown: network connection packages are collected and delivered through the Set 1 Data Preprocessing task, after which the modified data are transferred to Layer 1, which is a Shallow learning framework like Gaussian Naive

Bayes, to ascertain whether the connection is Normal, DoS, or Probe. The connection is extremely unlikely to be a Probe or DoS if the forecast is incorrect. The second layer is then turned on.

The Set 2 Data Preprocessing method is applied to the original data. After that, Layer 2, which is SVM, receives the altered data and decides whether the connection is U2R, R2L, or regular. This link is anticipated to be normal if the prediction is incorrect. If one of the two classifiers made a positive prediction, the link is cut off and designated as an abnormal connection. Due to the higher probability of DoS and Probe attacks, this framework is computationally effective in detecting Probe and DoS, followed by U2R and R2L. Algorithm 1 explains the TLCA algorithm.

Input: $X = \{f_1, f_2, \ldots, f_{40}\}$ // **40 features of input network traffic are captured**
Output: $y_o \in \{0, 1\}$
while TLCA Intrusion Detection System under execution **do**
  // for each network connection following the completion of Set 1 data pre-processing depict X_i as X_{p1}
if *Layer*1 anticipates X_{p1} as 1 **then**
  y_o = 1, **return** value of y_o
otherwise
  The second layer is turned on.
  following the completion of Set 2 data pre-processing depict X_i as X_{p2}
  if *Layer*2 anticipates X_{p2} as 1 **then**
    y_o = 1, **return** value of y_o
otherwise
  y_o = 0, **return** value of y_o
  end if
 end if
end while

There are not many ongoing operating expenses as a trade-off because our hybrid 2-classifier technique is focused on increasing the detection rates of U2R and R2L attacks. First, as the process of making decisions gets increasingly intricate and two adverse predictions are needed to demonstrate that the network connection is secure, the time dedicated to detecting attacks grows; furthermore, each layer's data pre-processing results in a greater resource requirement. To prevent traffic jams, strong equipment is advised for this method. Notably, methods for machine learning rely on high-quality data to build a robust model. A long-term IDS solution might benefit from collecting attack signatures, such as utilizing a honeypot method [31].

3. Results and Discussion

Table 3 shows the computer specifications used for experimental evaluations. To examine the framework on a big sample size, we ran experiments using training data sets KDDTrain+. The outcomes of these investigations enabled us to evaluate the functionality of our proposed TLCA. KDDTest+ data set is used for testing purpose.

Table 3. Computer Specifications used for conduction of experimental evaluations

Component	Specification
Operating System	Ubuntu 20.04 LTS
RAM	8 GB
Processor	Intel(R) Core (TM) i5-10210U CPU @ 1.60GHz
Processor Speed	2112 MHz
Number of Cores	4 Cores
Logical Processors	8 Logical Processors

We divided the training data into the two sets as described earlier at the training stage. Next, we ran the CCFS. The attributes among DoS and the rest that are connected $\{A_1, A_2, \ldots A_p\}$ include

$[A_8, A_{12}, A_{23}, A_{25}, A_{26}, A_{27}, A_{28}, A_{29}, A_{30}, A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36}, A_{37}, A_{38}, A_{39}, A_{40}, A_{41}]$. The shared attributes among Probe and the others $\{A_1, A_2, \ldots A_q\}$ = $[A_1, A_{12}, A_{23}, A_{24}, A_{25}, A_{27}, A_{28}, A_{29}, A_{30}, A_{31}, A_{33}, A_{34}, A_{35}, A_{36}, A_{37}, A_{38}, A_{39}, A_{40}, A_{41}]$. As a result, DoS and Probe's intersect attributes are $[A_{12}, A_{23}, A_{25}, A_{27}, A_{28}, A_{29}, A_{30}, A_{31}, A_{33}, A_{34}, A_{35}, A_{36}, A_{37}, A_{38}, A_{39}, A_{40}, A_{41}]$. The linked attributes of Normal and R2L $\{A_1, A_2, \ldots A_r\}$ in Set 2 are $[A_1, A_5, A_6, A_9, A_{10}, A_{11}, A_{12}, A_{14}, A_{18}, A_{22}, A_{23}, A_{24}, A_{29}, A_{30}, A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36}, A_{37}, A_{38}, A_{39}]$. The shared attributes among Normal and U2R $\{A_1, A_2, \ldots A_s\}$ are $[A_9, A_{10}, A_{12}, A_{14}, A_{17}, A_{18}, A_{24}, A_{31}, A_{32}, A_{33}, A_{36}, A_{37}]$. As a result, R2L and U2R intersect attributes are $[A_9, A_{10}, A_{12}, A_{14}, A_{18}, A_{24}, A_{31}, A_{32}, A_{33}, A_{36}, A_{37}]$. This is appropriate because, for example, count is frequently large in Probe and DoS assaults, and num_shell is frequently appropriate to U2R and R2L trends.

Min-max Normalization, as well as one-hot encoding, were then carried out, each in order. The final stage of the Data Pre-processing procedure is dimensionality reduction using PCA and KPCA. The threshold was set at 95% of the total variance. It suggested that Set 1, which accounted for 95 percent of the variation, has a reasonable number of 28 components. In Set 2, which accounted for 95% of the variance, 13 components were chosen. The often records, i.e., Normal on Set 2, were then down-sampled to maintain a 1:1 ratio between unusual and normal. In the final stage, Set 2's hyperparameters were tuned using a combination of linear and RBF kernel specifications.

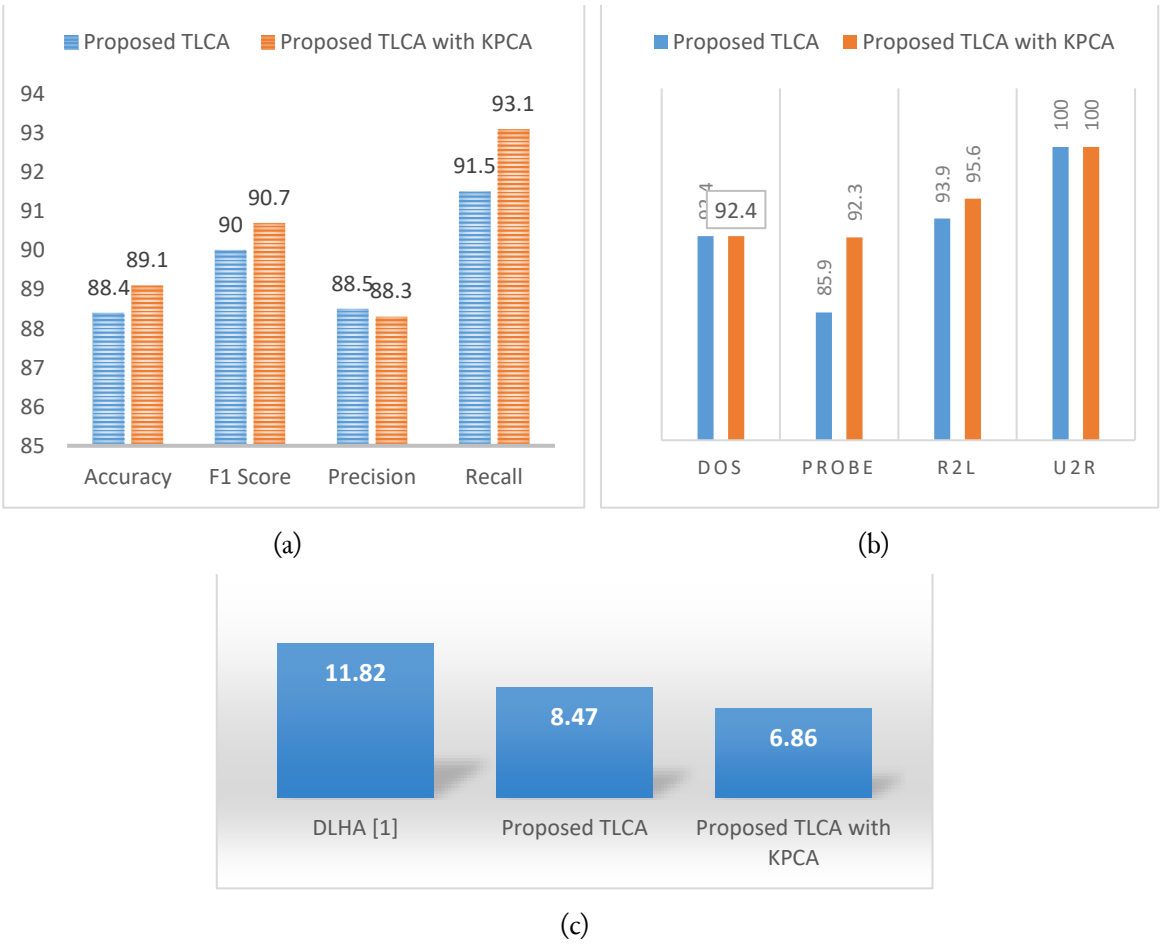

Fig. 5. (a)Performance of Proposed system. (b) Detection rates of main attack types. (c) False Alarm Rates of Proposed Models and DLHA

In the first experiment, training was done using KDDTrain+. To differentiate R2L and U2R attacks from another types of attacks, we tried to choose the most appropriate criteria. For the majority of

parameter combinations, the RBF kernel performed significantly better. With one exception, the SVM performs much worse when C is equivalent to 0.1 and gamma is equivalent to 10. Evidently, the detection rate decreased as the value of gamma increased, while C was equivalent to 0.1. Furthermore, the results are comparatively constant when C is equal to or larger than 1. The configuration with the highest detection rate is when the hyperparameter C value is equivalent to 0.1 and the gamma value is equivalent to 0.1. It achieved a respectable 0.9507 average detection rate and 0.0687 false alarm rate.

We evaluated TLCA using an unknown dataset, i.e., KDDTest+, utilizing a technique described in Algorithm 1 to evaluate our system on the two different experiments. With the help of KDDTrain+ during training, our suggested framework demonstrated excellent classification results as shown in Fig. 5(a) when C and gamma values are 0.1, attaining 88.4% accuracy, 90.0% F1 score, 88.5% precision, and 91.5% detection rate with 8.47% false alarm rate. The framework was also shown to be effective when PCA and Naive Bayes Classifier were used at layer1 and Kernel PCA and SVM with rbf kernel at layer 2, where it achieved improved results like: accuracy of 89.1%, F1 score of 90.7%, precision of 88.3%, and detection rate of 93.1% with a false alarm rate of 6.86%.

We next performed a thorough evaluation of our findings to investigate the rate of detection for every category, as demonstrated in Fig. 5(b). The detection rates for our suggested method when C and gamma values are 0.1 were found to be 92.4% for DoS (6,893 of 7,460), 85.9% for Probe (2080 of 2,421), 93.9% for R2L (2,709 of 2,885), and 100% for U2R (67 of 67) while utilizing KDDTrain+ during the training phase. When utilizing Kernel PCA for Set 2 dimensionality reduction, it possesses detection percentages of 92.4% for DoS (6,893 of 7,460), 92.3% for Probe (2,235 of 2,421), 95.6% for R2L (2,758 of 2,885), and 100% for U2R (67 of 67). Thus, our suggested TLCA succeeded in achieving its goal of maintaining high detection rates on Probe and DoS and excelling in detecting 100% on U2R and 95.6% on R2L in KDDTest+. This strong performance aligns with the rationale behind the dataset partitioning strategy. A 2D PCA projection (Fig. 5(b)) revealed that DoS and Probe instances formed well-defined, separable clusters, whereas R2L, U2R, and Normal samples exhibited substantial overlap in reduced-dimensional space. This observation supported the decision to split the data into Set 1 (all classes) and Set 2 (Normal, R2L, U2R), enabling the use of a secondary classifier optimized for resolving nuanced, low-frequency attack types. The clustering patterns were further corroborated by statistical analysis of intra- and inter-class Euclidean distances within the PCA space.

The ability of our method to identify more sorts of attacks in KDDTest+, the attack types which are not present in the training set, is one of the most crucial aspects of this study that we have highlighted. In KDDTest+, there are 12,833 attacks; 9,083 of them fall under recognized attack categories, while 3,750 fall under unknown attack categories. TLCA obtained detection rates of 92.4% (8,393 of 9,083) for recognized attack types and 89.4% (3,353 of 3,750) for unknown attack types while training with KDDTrain+. While using TLCA with KPCA obtained rates of detection 94.1% (8,547 of 9,083) for recognized attack types and 90.8% (3,405 of 3,750) for unknown attack types.

According to the findings, TLCA did a remarkable job of recognizing both known and new attack kinds. Due to the larger number of samples in KDDTrain+ for each category, TLCA was able to recognize 94.01% of attacks through recognized attack types while the entire data was utilized for training. Fig. 5(c) presents the False Alarm Rate (FAR) performance of the proposed TLCA framework, including its KPCA-enhanced variant, across a broader set of competing methods consistent with those listed in Table 4 and Table 5. TLCA achieves FAR values of 8.47 and 6.86 when integrated with KPCA—both improvements over the baseline DLHA model. To ensure a fair comparison, we trained exclusively on KDDTrain+ and evaluated on KDDTest+, and thus only included prior works that also used KDDTest+ for performance reporting. This expanded comparison highlights TLCA's ability to maintain high detection accuracy while significantly reducing false positives, an essential balance in practical IDS deployments.

We conducted a thorough contrast of our findings with other IDS research articles that were publicly available to impartially assess our outlined framework on broader consequences, as shown in Table 4.

Table 4. Performance comparison to alternative anomaly-based IDS techniques based on Detection rate, Precision, F1 score, Accuracy, and False Alarm Rate (only compared with research that conducted assessment in the authentic KDDTest+).

Sr. No.	System/Algorithm	Year	Training Dataset	Detection Rate (%)	Precision (%)	F1 Score (%)	Accuracy (%)	FAR (%)
1	VAE-WGAN [39]	2024	KDDTrain+	83.45	84.62	83.69	83.45	NA
2	DLHA [1]	2022	KDDTrain+	93.11	88.17	90.57	88.97	11.82
3	DLHA [1]	2022	KDDTrain+_20%	90.24	88.16	89.19	87.55	11.83
4	Stacked Ensemble-IDS [24]	2022	KDDTrain+	77	78.75	77.25	81.67	NA
5	CRNN [2]	2023	KDDTrain+	NA	NA	NA	84.30	NA
6	Kasongo Simple RNN [42]	2023	KDDTrain+	NA	NA	NA	83.70	NA
7	LSTM [42]	2023	KDDTrain+	NA	NA	NA	88.13	NA
8	GRU [42]	2023	KDDTrain+	NA	NA	NA	85.70	NA
9	Combined K-Means +RF [39]	2021	KDDTrain+	NA	NA	NA	85.24	NA
10	CNN + BiLSTM [17]	2020	KDDTrain+	84.49	85.82	85.14	83.58	NA
11	Autoencoder [33]	2020	KDDTrain+	80.37	87	81.98	84.24	NA
12	TSE-IDS [14]	2019	KDDTrain+	86.8	88.0	NA	85.797	117
13	Adaptable Ensemble [7]	2019	KDDTrain+	86.5	86.5	84.9	85.2	NA
14	Sparse AE + SVM [28]	2018	KDDTrain+	76.57	96.23	85.28	84.96	NA
15	TDTC [5]	2016	KDDTrain+	84.86	NA	NA	NA	4.86
16	PSOM+PCA+FDR [32]	2015	KDDTrain+	92.0	NA	NA	88.0	NA
17	**Proposed TLCA**	**2025**	**KDDTrain+**	**91.5**	**88.5**	**90**	**88.4**	**8.47**
18	**Proposed TLCA with KPCA**	**2025**	**KDDTrain+**	**93.1**	**88.3**	**90.7**	**89.1**	**6.86**

Our framework is recognized as one of the best in the industry. Evidently, TLCA achieves the greatest DR and F1 Score. The results are extrapolated from the original NSL–KDD publication and used as a benchmark. Substandard models are those that perform below the baseline. Our TLCA offers a +7.08% accuracy advantage over the optimal standard one machine learning classifier, NB Tree, and a +11.69% advantage over Multi-Layer Perceptron. Additionally, SVM and RNN, a one machine learning classifier framework designed to find all sorts of attacks, were developed by [20], [35]. They received accuracy ratings of 82.37% and 81.29%, in that order, showing no better than the baseline, whereas the maximum of hybrid techniques outperformed it.

Additionally, as shown in Table 5, we compared our rates of detection of the main attack groups to those of another research. Comparing the findings shows that while some achieve better results, TLCA is not the greatest method for detecting DoS or Probes. Our results fall between 85% and 92%. In contrast to existing models that show unfavorable detection ratings for R2L and U2R, our model has the capacity to reliably identify all sorts of attacks. With detection rates of 100% in U2R and 95.6% in R2L, our model significantly surpasses all competing approaches.

Table 4 and Table 5 provide a detailed performance comparison between the proposed TLCA framework and several state-of-the-art IDS models. These results confirm that TLCA achieves notably higher detection rates for R2L and U2R attack categories, often underrepresented in training data and poorly detected by many existing approaches. Furthermore, TLCA demonstrates improved generalization to unknown attack types, validating its robustness. All models in the comparison were

evaluated under consistent preprocessing conditions using the NSL-KDD dataset to ensure fairness in benchmarking.

Table 5. Comparison of detection rates of main attack types with existing studies

Sr. No.	System/Algorithm	DoS	Probe	R2L	U2R	Known Attacks	Unknown Attacks
1	DLHA [1]	92.4	90.87	96.67	100	94.01	90.9
2	Stacked Ensemble-IDS [24]	95	92	45	76	NA	NA
3	TDTC [30]	88.20	87.32	42	70.15	NA	NA
4	Combined KMeans+RF [39]	90.42	91.53	73.84	25.79	NA	NA
5	Dual-Tier [5]	84.68	79.76	34.81	67.16	NA	NA
6	PCA and KNN [40]	94.23	78.86	69.87	80.09	NA	NA
7	SVM and BIRCH [41]	97.5	99.5	19.7	28.8	NA	NA
8	Dual-Level [48]	97.37	94.72	14.02	90.71	NA	NA
9	Adaptive Ensemble [7]	84.37	87.11	55.27	25.0	NA	NA
10	**Proposed TLCA**	**92.4**	**85.9**	**93.9**	**100**	**93.5**	**91.1**
11	**Proposed TLCA with KPCA**	**92.4**	**92.3**	**95.6**	**100**	**94.1**	**90.8**

TLCA's improved performance in detecting R2L and U2R attacks can be attributed to key design decisions in Layer 2. The use of KPCA enables the extraction of nonlinear principal components, thereby capturing the subtle patterns and structural variations in complex, low-frequency classes more effectively. This enhances feature separability, which is critical for R2L and U2R instances that often overlap with normal traffic in linear space. Furthermore, employing an SVM with an RBF kernel in this layer provides a robust nonlinear decision boundary, effectively modeling the intricate distributions of these rare attack categories. This combination enables TLCA to generalize better to previously unseen and ambiguous samples, which is reflected in its superior detection rates.

4. Conclusion

IDS approaches based on rules are inadequate in the current period of expanding global internet connectivity. To address inadequate performance in infrequent attacks, this paper proposes a technique called the Two-Layered Collaborative Approach (TLCA), which also leads to an enhanced overall detection rate. To decrease the measures and speed up the overall structure for in-the-moment exercise, a Cross-sectional Correlated Feature Selected (CCFS) was introduced as a component of TLCA to reject frequently unnecessary features on the subgroups. Also, PCA and Kernel PCA are used for further reducing the dimensionality and selecting the principal components from the Set 1 and Set 2 datasets. There are two levels in the detection section. To categorize Probe and DoS attacks across all connections, the initial level utilized Gaussian Naïve Bayes as it gives better outcomes. To detect U2R and R2L attacks within regular traffic, which is a more challenging task, the second level used SVM with an RBF kernel. SVM optimization was done for c and gamma, as they are the major parameters to correctly recognize attacks that adhere to a comparable design to regular connections, i.e., U2R and R2L. Hyperparameter adjustment is crucial. The NSL-KDD dataset was utilized to assess our suggested TLCA. With a total rate of detection of 93.11%, an R2L detection rate of over 95.6%, and a U2R detection rate of 100%, it produced remarkable results. Also proposed TLCA gives improved detection rate results for known attacks, it is 94.1% and unknown attacks, it is 91.1% while using the KDDTest+ dataset for testing. Its improved efficiency and capacity for larger applications have been demonstrated by the execution time and F1 score. The outcomes of our experiments demonstrated the effectiveness and efficiency of the combined IDS strategy when two distinct classifiers are combined with CCFS. Since we used hyperparameter tuning utilizing training data to avoid overfitting and data leakage, we accomplished our goal of creating a generalized model with the best accuracy in spotting unusual but more severe threats. This strategy tries to secure crucial network settings and is appropriate for a real-time IDS. Future work on this subject may involve applying this methodology to other data sets or network environments that categorize attacks in different ways, such as when there are more than four different sorts of attacks.

Declarations

Author contribution. Both authors contributed to the writing, review, and revision of the manuscript. All authors reviewed the manuscript.
Funding statement. Not Applicable
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

Data and Software Availability Statements

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

References

[1] T. Wisanwanichthan and M. Thammawichai, "A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM," *IEEE Access*, vol. 9, pp. 138432–138450, 2021, doi: 10.1109/ACCESS.2021.3118573.

[2] M. H. and S. Kumar, "Network Intrusion Detection System using Convolution Recurrent Neural Networks and NSL-KDD Dataset," *Fusion Pract. Appl.*, vol. 13, no. 1, pp. 117–125, 2023, doi: 10.54216/FPA.130109.

[3] 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)*, Sep. 2020, pp. 149–155, doi: 10.1109/ICOSEC49089.2020.9215333.

[4] 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*, Jul. 2009, no. July, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[5] 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 Trans. Emerg. Top. Comput.*, vol. 7, no. 2, pp. 314–323, Apr. 2019, doi: 10.1109/TETC.2016.2633228.

[6] 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. 33789–33795, May 2018, doi: 10.1109/ACCESS.2018.2841987.

[7] 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. 82512–82521, 2019, doi: 10.1109/ACCESS.2019.2923640.

[8] 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, Mar. 2019, doi: 10.3390/en12071223.

[9] 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)*, Oct. 2019, pp. 1–6, doi: 10.1109/WCSP.2019.8927907.

[10] 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)*, Nov. 2017, vol. 2017-Novem, pp. 38–41, doi: 10.1109/ICSESS.2017.8342859.

[11] 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," *J. Electr. Comput. Eng.*, vol. 2014, no. 1, pp. 1–8, Jan. 2014, doi: 10.1155/2014/240217.

[12] 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)*, Jun. 2016, pp. 104–107, doi: 10.1109/TSP.2016.7760838.

[13] 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)*, Sep. 2018, pp. 2441–2446, doi: 10.1109/ICACCI.2018.8554710.

[14] 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. 94497–94507, 2019, doi: 10.1109/ACCESS.2019.2928048.

[15] W. Feng, Q. Zhang, G. Hu, and J. X. Huang, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Futur. Gener. Comput. Syst.*, vol. 37, pp. 127–140, Jul. 2014, doi: 10.1016/j.future.2013.06.027.

[16] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *J. Inf. Secur. Appl.*, vol. 44, pp. 80–88, Feb. 2019, doi: 10.1016/j.jisa.2018.11.007.

[17] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network," *IEEE Access*, vol. 8, pp. 32464–32476, 2020, doi: 10.1109/ACCESS.2020.2973730.

[18] Y. C. C. Mohammadpour Leila, Chaw Ling Teck, Sun Liew Chee, "A Convolutional Neural Network for Network," *A Convolutional Neural Netw. Netw. Intrusion Detect. Syst.*, pp. 50–55, 2018, [Online]. Available at: https://core.ac.uk/download/pdf/229876031.pdf.

[19] 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*, Dec. 2018, pp. 81–85, doi: 10.1145/3297156.3297230.

[20] 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. 21954–21961, Oct. 2017, doi: 10.1109/ACCESS.2017.2762418.

[21] H. Yao, Q. Wang, L. Wang, P. Zhang, M. Li, and Y. Liu, "An Intrusion Detection Framework Based on Hybrid Multi-Level Data Mining," *Int. J. Parallel Program.*, vol. 47, no. 4, pp. 740–758, Aug. 2019, doi: 10.1007/s10766-017-0537-7.

[22] 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 *Communications in Computer and Information Science*, vol. 1076, Springer, Singapore, 2019, pp. 274–287, doi: 10.1007/978-981-15-0111-1_25.

[23] B. S. Sukhadeo, R. N. Patil, R. Atole, Y. D. Sinkar, U. C. Patkar, and R. Chopade, "MLIDS: A Machine Learning-Based Intrusion Detection System Using the NSLKDD Data," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 4s, pp. 167–179, Nov. 2023. [Online]. Available at: https://www.ijisae.org/index.php/IJISAE/article/view/3761.

[24] O. Article, "Stacked Ensemble-IDS Using NSL-KDD Dataset," *J. Pharm. Negat. Results*, vol. 13, no. SO3, pp. 351–356, Jan. 2022, doi: 10.47750/pnr.2022.13.S03.057.

[25] A. H. Ali *et al.*, "Unveiling machine learning strategies and considerations in intrusion detection systems: a comprehensive survey," *Front. Comput. Sci.*, vol. 6, p. 1387354, Jun. 2024, doi: 10.3389/fcomp.2024.1387354.

[26] E. U. H. Qazi, M. H. Faheem, and T. Zia, "HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System," *Appl. Sci.*, vol. 13, no. 8, p. 4921, Apr. 2023, doi: 10.3390/app13084921.

[27] Y. Almutairi, B. Alhazmi, and A. Munshi, "Network Intrusion Detection Using Machine Learning Techniques," *Adv. Sci. Technol. Res. J.*, vol. 16, no. 3, pp. 193–206, Jul. 2022, doi: 10.12913/22998624/149934.

[28] G. Li, Z. Yan, Y. Fu, and H. Chen, "Data Fusion for Network Intrusion Detection: A Review," *Secur. Commun. Networks*, vol. 2018, no. 1, pp. 1–16, Jan. 2018, doi: 10.1155/2018/8210614.

[29] B. M. Aslahi-Shahri *et al.*, "A hybrid method consisting of GA and SVM for intrusion detection system," *Neural Comput. Appl.*, vol. 27, no. 6, pp. 1669–1676, Aug. 2016, doi: 10.1007/s00521-015-1964-2.

[30] H. H. Pajouh, G. Dastghaibyfard, and S. Hashemi, "Two-tier network anomaly detection model: a machine learning approach," *J. Intell. Inf. Syst.*, vol. 48, no. 1, pp. 61–74, Feb. 2017, doi: 10.1007/s10844-015-0388-x.

[31] M. Baykara and R. Das, "A novel honeypot based security approach for real-time intrusion detection and prevention systems," *J. Inf. Secur. Appl.*, vol. 41, pp. 103–116, Aug. 2018, doi: 10.1016/j.jisa.2018.06.004.

[32] 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, Sep. 2015, doi: 10.1016/j.neucom.2014.09.083.

[33] 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, doi: 10.1016/j.neucom.2019.11.016.

[34] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, "MLH-IDS: A Multi-Level Hybrid Intrusion Detection Method," *Comput. J.*, vol. 57, no. 4, pp. 602–623, Apr. 2014, doi: 10.1093/comjnl/bxt044.

[35] 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)*, Dec. 2014, pp. 1–6, doi: 10.1109/SKIMA.2014.7083539.

[36] T. D. Diwan, S. Choubey, and H. S. Hota, "A Detailed Analysis on NSL-KDD Dataset using various Machine Learning Techniques for Intrusion Detection," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 11, pp. 2954–2968, 2021, [Online]. Available at: https://d1wqtxts1xzle7.cloudfront.net/96437866/a-detailed-analysis-on-nsl-kdd-dataset-using-various-machine-learning-techniques-for-intrusion-detection-libre.pdf?1672161868=&response-content-.

[37] 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)*, Dec. 2015, pp. 317–322, doi: 10.1109/ICCITechn.2015.7488089.

[38] N. K. Kanakarajan and K. Muniasamy, "Improving the Accuracy of Intrusion Detection Using GAR-Forest with Feature Selection," in *Advances in Intelligent Systems and Computing*, vol. 404, Springer, New Delhi, 2016, pp. 539–547, doi: 10.1007/978-81-322-2695-6_45.

[39] 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. 75729–75740, 2021, doi: 10.1109/ACCESS.2021.3082147.

[40] 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)*, Oct. 2018, pp. 1–6, doi: 10.1109/WINCOM.2018.8629718.

[41] S.-J. Horng *et al.*, "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 306–313, Jan. 2011, doi: 10.1016/j.eswa.2010.06.066.

[42] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Comput. Commun.*, vol. 199, pp. 113–125, Feb. 2023, doi: 10.1016/j.comcom.2022.12.010.

[43] R. Ben Said, Z. Sabir, and I. Askerzade, "CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection," *IEEE Access*, vol. 11, pp. 138732–138747, 2023, doi: 10.1109/ACCESS.2023.3340142.

[44] V. Hnamte and J. Hussain, "DCNNBiLSTM: An Efficient Hybrid Deep Learning-Based Intrusion Detection System," *Telemat. Informatics Reports*, vol. 10, p. 100053, Jun. 2023, doi: 10.1016/j.teler.2023.100053.

[45] M. Sajid *et al.*, "Enhancing intrusion detection: a hybrid machine and deep learning approach," *J. Cloud Comput.*, vol. 13, no. 1, p. 123, Jul. 2024, doi: 10.1186/s13677-024-00685-x.

[46] Z. Li, C. Huang, and W. Qiu, "An intrusion detection method combining variational auto-encoder and generative adversarial networks," *Comput. Networks*, vol. 253, p. 110724, Nov. 2024, doi: 10.1016/j.comnet.2024.110724.

[47] C. Rajathi and P. Rukmani, "Hybrid Learning Model for intrusion detection system: A combination of parametric and non-parametric classifiers," *Alexandria Eng. J.*, vol. 112, pp. 384–396, Jan. 2025, doi: 10.1016/j.aej.2024.10.101.

[48] C. Guo, Y. Ping, N. Liu, and S.-S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, Nov. 2016, doi: 10.1016/j.neucom.2016.06.021.