A Hybrid Intrusion Detection System Combining PCAMIX, KPCA, and Random Forest for Enhanced Anomaly Detection

ABSTRACT

.

|  |
| --- |
| This study aims to develop a hybrid intrusion detection system (PKRIDS) that integrates PCAMIX-based Hotelling's T² control charts, Kernel Principal Component Analysis (KPCA), and Random Forest (RF) to improve detection accuracy while reducing false positives in network security. The hybrid approach combines statistical process control, nonlinear dimensionality reduction, and machine learning techniques. Evaluation on benchmark datasets NSL\_KDD and TON\_IoT used metrics including accuracy, precision, recall, F1-score, and ROC-AUC. PKRIDS employs PCAMIX for mixed-type data processing, KPCA for nonlinear pattern recognition, and RF for robust classification. On NSL\_KDD, the model achieved 99.81% detection rate with 0.18% false positives (ROC-AUC=0.9975). For TON\_IoT, it attained 99.86% detection rate with 0.13% false positives (ROC-AUC=0.9975). These results demonstrate PKRIDS's effectiveness in combining statistical and machine learning methods for enhanced intrusion detection. The system shows particular strength in handling both continuous and categorical variables while maintaining low false alarm rates. |

*Keywords: Intrusion Detection System (IDS), PCAMIX, KPCA, Random Forest, Hybrid Model, Anomaly Detection, Network Security.)*

1. INTRODUCTION

Intrusion is a formal term describing the act of compromising system resources. An intrusion can be defined as "any set of actions that attempt to compromise the integrity, confidentiality, or availability of a system resource". Detecting either failed or successful attempts to compromise a system is known as Intrusion Detection. Intrusion detection systems (IDS) detect possible intrusions. The goal of IDS tools is to detect computer attacks or illegal access and to alert concerned people about detection or security breaches (kamini, 2020). An intrusion compromises the security (e.g. availability, integrity, and confidentiality) of an information system through various means, including denial-of-service, remote-to-local, user-to-root, and information probing (Rajkumar et al, 2015).

The rapid growth of internet-based services has led to an increase in cyberattacks, making network security a critical concern. Intrusion Detection Systems (IDS) are essential tools for identifying and mitigating these threats. However, traditional IDS methods, such as signature-based and anomaly-based detection, face challenges in handling high-dimensional and mixed-type data, often resulting in high false positive rates and inconsistent performance (Axelsson, 2000; Kabiri et al., 2005).

Signature-based IDS rely on predefined patterns of known attacks, making them ineffective against novel or zero-day attacks (Ahmim et al, 2018). On the other hand, anomaly-based IDS detect deviations from normal behavior but often suffer from high false positive rates due to the difficulty in defining a clear boundary between normal and abnormal activities (Chandola et al., 2009). Furthermore, traditional IDS methods struggle with the complexity of network intrusion data, which typically includes a mixture of continuous and categorical variables with nonlinear relationships (Tang et al, 2020).

Wang et al. (2023) proposed a hybrid IDS that leverages both a random forest (RF) and an autoencoder, they utilized the probability output of the RF classifier to determine whether a sample belongs to an attack. Unknown attacks can be identified using a probability output. An additional AE was added to reduce the false positive rate. To simulate an unknown attack in the experiments, they explicitly removed samples belonging to one attack class from the training set. Furthermore, the additional AE detection module decreased the false positive rate.

Shaohui et al. (2021) proposed Hotelling's T2 multivariate control charts based on Principal Component Analysis mix (PCA mix) with a bootstrap control limit and applied it to the network intrusion detection system. It was compared with the conventional Hotelling's T2 control chart based on PCA. The experimental results revealed that the proposed method performed better than its counterparts in terms of intrusion detection.

In recent years, there has been a significant surge in interest regarding the application of anomaly detection methods that leverage multivariate statistical process control and machine learning techniques to identify security breaches within computer networks (Kamini, 2020; Shaohui, 2021). Despite the advancements in these methodologies, several critical challenges persist. Compounding these challenges is the complexity of network intrusion data, which typically exhibit a mixture of high-dimensional datasets that include both continuous and categorical variables as well as nonlinear relationships.

To address these limitations, this paper proposes a hybrid intrusion detection system (PKRIDS) that combines PCAMIX-based Hotelling's T² control charts, Kernel Principal Component Analysis (KPCA), and Random Forest (RF). The hybrid model leverages the strengths of statistical process control, nonlinear dimensionality reduction, and machine learning to improve detection accuracy and reduce false positives. The proposed approach is evaluated on two benchmark datasets, NSL\_KDD and TON\_IoT, demonstrating its effectiveness in detecting both known and unknown attacks while maintaining low false alarm rates. The hybrid model leverages the strengths of statistical process control, nonlinear dimensionality reduction, and machine learning to improve detection accuracy and reduce false positives.

The main contributions and findings of this paper are as follows:

* A hybrid intrusion detection system (PKRIDS) is proposed, which combined PCAMIX-based Hotelling's T² control charts, Kernel Principal Component Analysis (KPCA), and Random Forest (RF). It improves the accuracy of IDS and provides a new research method for intrusion detection.
* Scalability: The hybrid approach is scalable and can be applied to different datasets and environments.
* Robustness: The integration of statistical, nonlinear, and machine learning techniques ensures robust performance in detecting both known and unknown attacks.

2. material and methods

This section presents an intrusion detection method based on multivariate control charts. The establishment of the monitoring system is divided into two steps: data preprocessing and the development of a hybrid intrusion detection method.

**2.1 Datasets**

In this study, we used the well-known NSL\_KDD and TON\_IoT datasets.

**2.1.1 NSL\_KDD dataset**

NSL\_KDD is a dataset created to solve some of the problems in the KDDCUP99 dataset. This dataset adjusts the proportion of positive and negative samples of the KDD‐CUP99 dataset, makes the number of samples in the training and testing datasets more reasonable, and removes redundant data in the KDDCUP99 dataset. Although this may not be a perfect representation of existing real networks, it can be used as a benchmark dataset for different intrusion detection methods in the absence of a public intrusion detection dataset based on the network.

In this study, the kddtrain+ file processed using the original dataset was selected as the experimental dataset. The dataset is a mixture of high-dimensional datasets with 41 attributes, including 32 continuous attributes and nine discrete attributes. In addition, not only in the training dataset, the normal data are approximately 53.45%, the rest are attack data, the difference between attack data and normal data is unbalanced, but also in the attack data, and the data proportion of various types of attacks is also extremely unbalanced.

**Table 1: Characteristics of the NSL\_KDD dataset**

|  |  |  |
| --- | --- | --- |
| **Class** | **Size** | **Percent (%)** |
| Normal | 8449 | 55.6% |
| Dos | 5,226.05 | 34.4% |
| Probe | 1,390.068 | 9.15% |
| U2R | 116.97 | 0.77% |
| R2L | 12.15 | 0.08% |
| **Total** | **15,192** | **100** |
|  |  |  |

**2.1.2 TON\_IoT dataset**

The TON\_IoT dataset is a new Internet of Things (IoT) and Industrial IoT (IIoT) dataset collected from a realistic and large-scale network in 2020. The network is designed at the Cyber Range and IoT Labs at UNSW Canberra (Australia) to mimic the complexity and scalability of industrial IoT and Industry 4.0. The dataset contains normal data and 7 types of attack data: backdoor, ddos, ransomware, injection, xss, password, and scanning (Alsaedi *et al.* 2020). This work selects ‘Train\_Test\_Windows 7’ file as the experimental dataset. The dataset also belongs to a mixture of high-dimensional datasets with 132 attributes, including 120 continuous attributes and 12 discrete attributes. In addition, not only in the training dataset, the normal data are approximately 62.58%, the rest are attack data, the difference between attack data and normal data is unbalanced, but also in the attack data, and the data proportion of various types of attacks is also extremely unbalanced, as shown on Table 2.

**Table 2: Characteristics of the TON\_IoT dataset**

|  |  |  |
| --- | --- | --- |
| **Class** | **Size** | **Percent (%)** |
|  Normal | 5,680.39 | 62.58% |
| DDos | 1,211.78 | 13.35% |
|  Backdoor | 1,010.27 | 11.13% |
|  Injection | 567.31 | 6.25% |
|  Password | 430.25 | 4.74% |
|  Scanning | 127.99 | 1.41% |
|  Ransomware | 46.29 | 0.51% |
|  XSS | 2.72 | 0.03% |
| **Total** | **9,077** | **100** |
|  |  |  |

**2.2 Data Pre-processing**

Data preprocessing in this approach, prior to dimensional reduction, is divided into three major components. Feature extraction, converting eigenvalues, and managing missing values.

**2.2.1 Feature extraction:** By removing various features that have a large number of similar values or missing values and do not affect the final result, the NSL\_KDD dataset has 5 discrete and 13 continuous features, whereas the TON\_IoT dataset has 1 discrete and 42 continuous features. (Table 3).

**Table 3: Features of datasets**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Class** | **Num** | **Total** |
| NSL\_KDD |  Continuous | 13 | 18 |
|  | Discrete | 5 |  |
| TON\_IoT |  Continuous | 42 | 43 |
|  | Discrete | 1 |  |

**2.2.2 Transforming eigen values**

To facilitate the calculation, all features were transformed to numbers. Only discrete features must be transformed into character types in this phase, whereas continuous features must be standardized, because the PCAMIX approach can immediately reduce the dimension of high-dimensional and mixed

data.

**Table 4: Eigenvalue transformation**

|  |  |  |
| --- | --- | --- |
| Dataset | Feature’s name | Eigenvalue transformation |
| NSL\_KDD | Protocol type | UDP=1, TCP=2, ICMP=3 |
|  | Flag | RSTR = 1; S0 = 2; S1 = 3; S2 = 4; S3 = 5; SF = 6; SH = 7OTH = 8; REJ = 9; RSTO = 10; RSTOS0 = 11 |
|  | Class | Normal=0, Abnormal=1 |

**2.2.3** **Handling missing values**

The dataset will be examined during the data preparation process to determine whether any attribute values were missing. This process is carried out by removing features with a large number of the same or missing values and does not affect the final result.

**2.3 PCAMIX-based Hotelling's T² Control Chart**

The PCA mix algorithm is used to reduce the dimension of the data and perform generalized singular value decomposition (GSVD) for the data, and then, the multivariate Hotelling's T2 control chart is used to train the reduced dimension data. The PCAmix method involves merging standard PCA and Multiple Correspondence Analysis (MCA). Shaohui *et al*. (2021), as described below. The specific steps are as follows:

* Let *m* denote the number of observation units, $α\_{1}$ denote the number of continuous variables, $α\_{2}$ denote the number of discrete variables, $A\_{1}$ denote the *m* x $α\_{1}$ continuous matrix and $A\_{2}$ denote the *m* x $α\_{2}$ discrete matrix.
* Let *A* be the *m* x ($α\_{1+}α\_{2})$ matrix that combines the normalized matrix $A\_{1}$ and normalized matrix $A\_{2}$: *A=(* $A\_{1},A\_{2}$ *).*
* Build a diagonal matrix *B*, that is, the weights matrix of the rows of *A*. The *b* rows are weighted by $\frac{1}{b}$, such that *B=*$\frac{1}{b}$$I\_{b}$
* Build a diagonal matrix *C*, that is, the weights matrix of the columns of *A*. The $α\_{1}$ first columns are weighted by 1 (as the calculation method of the Euclidean distance in PCA). The $α\_{2}$ last columns are weighted by $\frac{b}{b\_{j}}$ (as the weighted distance represented by *X*2 distance in MCA), where j=1, 2, …,$α\_{2}$*.*

That is, C = diag (1, 1, …,$\frac{n}{n\_{1}}, ... ,\frac{n}{n\_{α\_{2}}}$).

* The GSVD of *A* gives the decomposition

*A = Y Λ* $Z^{T}$

where Λ is the diagonal matrix of the radical of the eigenvalue of matrix *A*. Let *r* represent the rank of matrix *A*; then, *Y* be the matrix of the eigenvector of the *m* x *r* matrix *A* and let *Z* be the matrix of the eigenvector of the ($α\_{1}+α\_{2}$) x *r* matrix *A*.

The eigenvector matrix *Y* can be expressed as follows:

*Ymix = A C Z*

* Or directly computed from the GSVD decomposition as

*Ymix*= *Y* Λ

* The eigenvector matrix *Z* can be expressed as

*Zmix* = *CZ*Λ

Where matrix *Zmix*  is divided up as follows: $Z^{mix}\_{1}$ contains the factor scores of the $α\_{1}$ continuous variable and $Z^{mix}\_{2}$ contains the factor score of the $α\_{2}$ discrete variables.

* Calculate the mean value *α* of matrix *Zmix* after dimension reduction and use *n* to

 represent the dimension of *Zmix :* $α=\frac{m\_{1}+m\_{2}+. . . +m\_{n}}{n}$

* The covariance matrix *ϕ* and the *T*2 statistic can be calculated as follows:

$ϕ=\frac{(m\_{1}- α)(m\_{1}-α)^{I}+(m\_{2}-α)(m\_{2}-α)^{I}+ . . . +(m\_{m}-α)(m\_{m}-α)^{I}}{m - 1}$

*T2 =*$m(m\_{i}-α)^{I}ϕ^{-1}(m\_{i}- α)$

**2.4 Kernel Principal Component Analysis (KPCA)**

PCA is the basis of transformation to diagonalize the estimated co-variance matrix 𝐂 from input data. PCA was originally proposed for linear data. Therefore, this method is ineffective for nonlinear data. To overcome this nonlinear problem, Schölkopf *et al*. (1997) proposed a Kernel PCA scheme.

The basic idea of Kernel PCA is calculating the Principal Component Scores in higher dimensional space by conducting a nonlinear mapping Φ ∶ℝ𝑝→𝐹, 𝑦 ↦𝐘. This mapping can be executed by utilizing kernel functions known from the Support Vector Method (SVM) (Boser *et al*., 1992). Assume that the centered data are mapped to feature space 𝐹, Φ ($y\_{1}$), ..., Φ ($y\_{n}$). The feature space covariance matrix with a size of 𝑛 × 𝑛.

$C^{F}=\frac{1}{n}\sum\_{j=1}^{n}ϕ(y\_{j})ϕ(y\_{j})^{T}$

The next step is estimating the eigenvalues 𝜆 ≥ 0.

𝜆𝐕 = 𝐂𝐹𝐕

In general, mapping Φ(.) is not always calculated. To solve this problem, a dot product calculation from to vector in the feature space is performed. Let 𝐊 with a size of 𝑛 × 𝑛 defined as $k\_{ij}$=$\left〈ϕ\left(y\_{i}\right),ϕ(y\_{j})\right〉$. The principal component score (PCs) $t$ is computed using projection of Φ ($y\_{i}$) to eigenvector $V\_{v}$, where 𝑣 =1, 2, ..., 𝑙.

$t\_{v}=\left〈V\_{v}, Φ(y)\right〉=\sum\_{j=1}^{n}α\_{i}^{v}\left〈Φ\left(y\_{j}\right), Φ(y)\right〉$

Nonlinear mapping is not required to solve the eigenvalue problem and principal component calculation. To replace this, the kernel function can be constructed

 $K(y\_{i},y$)= $\left〈Φ\left(y\_{i}\right), Φ(y)\right〉$.

* 1. **Random Forest (RF)**

A random forest is a classifier consisting of a collection of tree-structured classifiers

 **{**$h\left(X,θ\_{k}\right), k=1, . . .n$**}** where the {$θ\_{k}$} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input$X$***.*** Random Forest (RF) is a powerful ensemble classifier that combines bagging and feature randomness to train multiple Decision Trees. (Breiman, 2001). Decision Tree is a frequently used classification approach. It attempts to learn a set of if–then rules to classify the data.

 

Figure 1. Illustration of Random Forest.

**2.6 Hybrid Model (PKRIDS)**

The hybrid model integrates the outputs of PCAMIX and KPCA into the Random Forest classifier. The steps are as follows:

* + 1. **Feature selection**

A crucial step in machine learning is feature selection, which eliminates superfluous or irrelevant features in order to determine the most pertinent input variables. It improves the simplicity, accuracy, and efficiency of the model (Baker, 2009; Mitra, 2002; Miller, 2002; Almuallim, 1994).

In order to maximize model performance with fewer features, this study used the training dataset (11,850 observations) instead of the entire dataset. To rank attributes by importance, three feature selection techniques were used: correlation, gain ratio, and information gain.

A step-wise modelling approach was used to identify the best features: models were constructed by starting with the top four ranked attributes and gradually adding more until all 30 attributes were present. The most predictive characteristics for the study were found with the aid of this methodical assessment.

**2.6.1.1 Correlation**

Using the correlation algorithm, Table 5 shows the features ranked from highest to lowest in order of importance. According to the findings, the top four most important characteristics are: Protocol type, Srv\_count, Rerror\_rate, Srv\_rerror\_rate, On the other hand, the least significant characteristics found were: Dst\_host\_count, Number\_compromised, Src\_bytes, Warm. Prioritizing important variables for model optimization is aided by this ranking.

=== Attribute selection 10-fold cross-validation (stratified), seed: 1 ===

|  |  |  |
| --- | --- | --- |
| average merit | average rank | attribute |
| 0.428 +- 0.002 | 1 +- 0 | 2 protocol\_type |
| 0.371 +- 0.002 | 2 +- 0 | 12 srv\_count |
| 0.261 +- 0.001 | 3 +- 0 | 15 rerror\_rate |
| 0.256 +- 0.001 | 4 +- 0 | 16 srv\_rerror\_rate |
| 0.256 +- 0.001 | 5 +- 0 | 4 flags |
| 0.251 +- 0.001 | 6 +- 0 | 29 dst\_host\_srv\_rerror\_rate |
| 0.246 +- 0.002 | 7 +- 0 | 28 dst\_host\_rerror\_rate |
| 0.238 +- 0.003 | 8 +- 0 | 21 dst\_host\_srv\_count |
| 0.208 +- 0.002 | 9 +- 0 | 17 same\_srv\_rate |
| 0.199 +- 0.003 | 10 +- 0 | 22 dst\_host\_same\_srv\_rate |
| 0.166 +- 0.001 | 11 +- 0 | 27 dst\_host\_srv\_serror\_rate |
| 0.157 +- 0.002 | 12.4 +- 0.66 | 26 dst\_host\_serror\_rate |
| 0.156 +- 0.001 | 12.7 +- 0.46 | 14 srv\_serror\_rate |
| 0.155 +- 0.001 | 13.9 +- 0.3 | 13 serror\_rate |
| 0.142 +- 0.003 | 15 +- 0 | 24 dst\_host\_same\_src\_port\_rate |
| 0.135 +- 0.001 | 16.3 +- 0.46 | 18 diff\_srv\_rate |
| 0.134 +- 0.003 | 16.7 +- 0.46 | 11 counts |
| 0.131 +- 0.001 | 18 +- 0 | 3 service |
| 0.095 +- 0.004 | 19 +- 0 | 23 dst\_host\_diff\_srv\_rate |
| 0.083 +- 0.004 | 20.4 +- 0.49 | 9 logged\_in |
| 0.083 +- 0.002 | 20.6 +- 0.49 | 8 num\_failed\_logins |
| 0.074 +- 0.001 | 22 +- 0 | 19 srv\_diff\_host\_rate |
| 0.055 +- 0.003 | 23.6 +- 0.8 | 25 dst\_host\_srv\_diff\_host\_rate |
| 0.055 +- 0.004 | 23.9 +- 0.7 | 6 dst\_bytes |
| 0.051 +- 0.005 | 24.7 +- 0.9 | 1 duration |
| 0.045 +- 0.003 | 25.8 +- 0.4 | 20 dst\_host\_count |
| 0.029 +- 0.003 | 27.1 +- 0.3 | 10 num\_compromised |
| 0.007 +- 0.001 | 28.3 +- 0.46 | 5 src\_bytes |
| 0.007 +- 0.009 | 28.6 +- 0.66 | 7 hots |

Table 5: **Pearson Correlation ranked features from the most important to the least important**

**2.6.1.2 Gain Ratio**

The features ranked by the Gain Ratio are listed in Table 6. From the third column, the four best attributes are Protocol\_type, dst\_bytes, src\_bytes, and srv\_diff\_host\_rate, whereas the least four features are count, dst\_host\_srv\_diff\_host\_rate, dst\_host\_count, and logged\_in. The Gain Ratio algorithm shares the three best attributes: (protocol\_type,dst\_bytes, and src\_bytes) and the two least attributes (dst\_host\_count and logged\_in) with the Information Gain Algorithm as shown in Table 7.

=== Attribute selection 10-fold cross-validation (stratified), seed: 1 ===

|  |  |  |
| --- | --- | --- |
| average merit  | average rank  |  attribute |
|  0.168 +- 0.002  | 1 +- 0  | 2 protocol type |
|  0.115 +- 0.001  | 2 +- 0  | 6 dst\_bytes |
|  0.11 +- 0.001  | 3 +- 0  | 5 src\_bytes |
|  0.078 +- 0.001  | 4.5 +- 0.67  | 19 srv\_diff\_host\_rate |
|  0.074 +- 0.006  | 5.2 +- 1.25  | 16 srv\_rerror\_rate |
|  0.065 +- 0.005  | 6.2 +- 1.17  | 29 dst\_host\_srv\_rerror\_rate |
|  0.063 +- 0  | 7.1 +- 0.7  | 3 service |
|  0.062 +- 0.003  | 7.6 +- 1.74  | 12 srv\_count |
|  0.061 +- 0.001  | 8.9 +- 0.7  | 15 rerror\_rate |
|  0.061 +- 0  | 9.5 +- 0.5  | 4 flags |
|  0.056 +- 0.001  | 11.2 +- 0.4  | 18 diff\_srv\_rate |
|  0.055 +- 0.001  | 11.8 +- 0.4  | 24 dst\_host\_same\_src\_port\_rate |
|  0.049 +- 0.002  | 13.6 +- 1.02  | 13 serror\_rate |
|  0.049 +- 0.001  | 13.9 +- 0.7  | 1 duration |
|  0.047 +- 0.001  | 15 +- 0.63  | 7 hots |
|  0.047 +- 0.001  | 15.9 +- 1.22  | 17 same\_srv\_rate |
|  0.046 +- 0.001  | 16.7 +- 0.46  | 8 num\_failed\_logins |
|  0.041 +- 0.001  | 18.6 +- 0.8  | 27 dst\_host\_srv\_serror\_rate |
|  0.04 +- 0.001  | 19.1 +- 0.83  | 28 dst\_host\_rerror\_rate |
|  0.04 +- 0.001  | 19.2 +- 0.87  | 14 srv\_serror\_rate |
|  0.034 +- 0.001  | 21.5 +- 0.67  | 10 num\_compromised |
|  0.032 +- 0.002  | 22.4 +- 0.8  | 26 dst\_host\_serror\_rate |
|  0.028 +- 0.003  | 23.2 +- 0.6  | 21 dst\_host\_srv\_count |
|  0.026 +- 0.008  | 23.9 +- 2.21  | 22 dst\_host\_same\_srv\_rate |
|  0.024 +- 0  | 24.4 +- 0.49  | 23 dst\_host\_diff\_srv\_rate |
|  0.02 +- 0  | 25.6 +- 0.49  | 11 counts |
|  0.016 +- 0.001  | 27 +- 0  | 25 dst\_host\_srv\_diff\_host\_rate |
|  0.012 +- 0.001  | 28 +- 0  | 20 dst\_host\_count |
|  0.006 +- 0.001  | 29 +- 0  | 9 logged\_in |

Table 6: Gain Ratio ranked features from the most important to the least important

**2.6.1.3 Information Gain**

The Information Gain algorithm is used to rank features in Table 7, with the top four attributes being src\_bytes, dst\_bytes, service, and protocol\_type. Features like num\_failed\_logins, num\_compromised, logged\_in, and dst\_host\_count are the least significant. Notably, among the three algorithms (Correlation, Gain Ratio, and Information Gain), protocol\_type is the most important feature, whereas dst\_host\_count and num\_compromised are consistently among the least important.

=== Attribute selection 10-fold cross-validation (stratified), seed: 1 ===

|  |  |  |
| --- | --- | --- |
|  average merit | average rank  |  attribute |
|  0.402 +- 0.003 | 1 +- 0 | 5 src\_bytes |
|  0.33 +- 0.002 | 2 +- 0 | 6 dst\_bytes |
|  0.261 +- 0.002 | 3 +- 0 | 3 service |
|  0.182 +- 0.002 | 4 +- 0 | 2 protocol\_type |
|  0.137 +- 0.001 | 5 +- 0 |  12 srv\_count |
|  0.11 +- 0.001 | 6 +- 0 |  4 flags |
|  0.102 +- 0.002 | 7.1 +- 0.3 |  24 dst\_host\_same\_src\_port\_rate |
|  0.097 +- 0.001 | 7.9 +- 0.3 |  29 dst\_host\_srv\_rerror\_rate |
|  0.081 +- 0.001 | 9.5 +- 0.67 |  18 diff\_srv\_rate |
|  0.08 +- 0.002 | 9.8 +- 0.75 |  28 dst\_host\_rerror\_rate |
|  0.078 +- 0.001 |  10.7 +- 0.46 |  1 duration |
|  0.072 +- 0.002 |  12.3 +- 0.46 |  21 dst\_host\_srv\_count |
|  0.072 +- 0.001 |  12.7 +- 0.46 |  15 rerror\_rate |
|  0.068 +- 0.001 |  14 +- 0 |  16 srv\_rerror\_rate |
|  0.06 +- 0.001 |  15.5 +- 0.81 |  19 srv\_diff\_host\_rate |
|  0.059 +- 0.001 |  16.2 +- 0.98 |  17 same\_srv\_rate |
|  0.058 +- 0.001 |  17.2 +- 0.87 |  23 dst\_host\_diff\_srv\_rate |
|  0.053 +- 0.006 |  18.1 +- 1.45 |  22 dst\_host\_same\_srv\_rate |
|  0.055 +- 0.003 |  18.2 +- 0.98 |  11 counts |
|  0.046 +- 0.001 |  19.8 +- 0.4 |  13 serror\_rate |
|  0.041 +- 0.002 |  21.4 +- 0.49 |  26 dst\_host\_serror\_rate |
|  0.04 +- 0 |  21.6 +- 0.49 |  27 dst\_host\_srv\_serror\_rate |
|  0.032 +- 0.001 |  23 +- 0 |  14 srv\_serror\_rate |
|  0.022 +- 0 |  24 +- 0 |  7 hot |
|  0.014 +- 0.001 |  25 +- 0 |  25 dst\_host\_srv\_diff\_host\_rate |
|  0.011 +- 0 |  26 +- 0 |  8 num\_failed\_logins |
|  0.007 +- 0 |  27 +- 0 |  10 num\_compromised |
|  0.005 +- 0 |  28.1 +- 0.3 |  9 logged\_in |
|  0.004 +- 0 |  28.9 +- 0.3 |  20 dst\_host\_count |

Table 7: Information Gain ranked features from most important to least important.

In summary, Protocol\_type, src\_bytes, and dst\_bytes are the top features that are consistently ranked by multiple algorithms, indicating their significance for intrusion detection. On the other hand, num\_compromised, logged\_in, and dst\_host\_count were often among the least significant, indicating little effect on classification.

**2.6.2 Performance Evaluation for Selected Features**

In this study, the qualities ranked for each feature selection method were sequentially modelled to assess the performance of the algorithms. This is accomplished by first choosing the top four attributes from each algorithm and then adding the next rated attribute one after the other until all 30 attributes have been included.

**2.6.2.1 Correlation**

According to Table 8, the PKRIDS model had the lowest RMSE (0.1465) with 17 features and the highest ROC (0.993) with 25 features. However, the Correlation algorithm did not achieve the objective of feature selection, which is to minimize features while maximizing performance, because the optimal performance required 17 to 25 features, which is nearly equal to the full 30 features. As a result, this approach was judged inappropriate for feature selection in this investigation.

**Table 8: Performance of the correlation ranked Attributes on the PKRIDS model**.

|  |
| --- |
| **PKRIDS Model** |
| No. of Attributes | ROC | RMSE |
| 4 | 0.981 | 0.1886 |
| 5 | 0.911 | 0.2653 |
| 6 | 0.929 | 0.2447 |
| 7 | 0.948 | 0.2279 |
| 8 | 0.959 | 0.2109 |
| 9 | 0.961 | 0.2091 |
| 10 | 0.975 | 0.1941 |
| 11 | 0.975 | 0.1916 |
| 12 | 0.978 | 0.1884 |
| 13 | 0.978 |  0.1884 |
| 14 | 0.979 | 0.1885 |
| 15 | 0.977 | 0.1872 |
| 16 | 0.989 |  0.1867 |
| 17 | 0.991 | **0.1465** |
| 18 | 0.990 | 0.1467 |
| 19 | 0.986 | 0.1628 |
| 20 | 0.987 | 0.1627 |
| 21 | 0.987 | 0.1629 |
| 22 | 0.987 | 0.1629 |
| 23 | 0.987 | 0.1608 |
| 24 | 0.990 | 0.1508 |
| 25 | **0.993** | 0.1469 |
| 26 | 0.945 | 0.2459 |
| 27 | 0.944 | 0.2466 |
| 28 | 0.945 | 0.2459 |
| 29 | 0.944 | 0.2461 |

 **Best 0.993 0.1465**

**2.6.2.2 Gain Ratio**

The outcomes of sequential modeling with the Gain Ratio approach are shown in Table 9. The IDS model had the lowest RMSE (0.1242) with 26 features and the highest ROC (0.997) with 21 features. However, the Gain Ratio algorithm also failed to find a minimal yet effective feature subset because the optimal performance required 21–26 features, which is close to the total feature count. As a result, this study comes to the conclusion that the Gain Ratio is inappropriate for feature selection in this particular situation.

**Table 9: Performance of the Gain Ratio ranked Attributes on the PKRIDS model**

|  |
| --- |
| **PKRIDS Model** |
| No. of Attributes | ROC | RMSE |
| 4 | 0.981 | 0.1694 |
| 5 | 0.969 | 0.1818 |
| 6 | 0.972 | 0.1802 |
| 7 | 0.980 | 0.1709 |
| 8 | 0.983 |  0.1502 |
| 9 | 0.978 | 0.1573 |
| 10 | 0.987 | 0.1429 |
| 11 | 0.987 | 0.1417 |
| 12 | 0.984 | 0.1468 |
| 13 | 0.979 | 0.1532 |
| 14 | 0.974 | 0.1568 |
| 15 | 0.981 | 0.1258 |
| 16 | 0.974 |  0.1550 |
| 17 | 0.975 | 0.1560 |
| 18 | 0.974 | 0.1549 |
| 19 | 0.971 | 0.1575 |
| 20 | 0.971 | 0.1575 |
| 21 | **0.997** | 0.1250 |
| 22 | 0.997 | 0.1297 |
| 23 | 0.997 | 0.1308 |
| 24 | 0.997 | 0.1308 |
| 25 | 0.997 | 0.1294 |
| 26 | 0.997 | **0.1242** |
| 27 | 0.997 | 0.1301 |
| 28 | 0.997 | 0.1315 |
| 29 | 0.997 | 0.1308 |

 **Best 0.997 0.1242**

**2.6.2.3 Information Gain**

The Information Gain method's sequential modeling results are displayed in table 10. With 17 features, the IDS model had the highest ROC (0.998), and with 18 features, the lowest RMSE (0.1288). In contrast to the other algorithms, Information Gain only needed 17–18 features to achieve optimal results, which is a substantial reduction from the full set of 30 features. This is the only appropriate algorithm for this purpose in the study since it effectively satisfies the feature selection objective of optimizing performance while minimizing features.

**Table 10: Performance of the Information Gain ranked Attributes on the PKRIDS model**

|  |
| --- |
| **PKRIDS Model** |
| No. of Attributes | ROC | RMSE |
| 4 | 0.953 | 0.1869 |
| 5 | 0.978 | 0.1993 |
| 6 | 0.988 | 0.1405 |
| 7 | 0.986 |  0.1404 |
| 8 | 0.984 |  0.1420 |
| 9 | 0.982 | 0.1441 |
| 10 | 0.978 | 0.1509 |
| 11 | 0.978 | 0.1509 |
| 12 | 0.988 | 0.1587 |
| 13 | 0.971 | 0.1585 |
| 14 | 0.971 | 0.1585 |
| 15 | 0.970 | 0.1583 |
| 16 | 0.971 |  0.1564 |
| 17 | **0.998** | 0.1299 |
| 18 | 0.997 | **0.1288** |
| 19 | 0.968 | 0.1541 |
| 20 | 0.968 | 0.1542 |
| 21 | 0.970 | 0.1535 |
| 22 | 0.975 | 0.1537 |
| 23 | 0.978 | 0.1540 |
| 24 | 0.969 | 0.1556 |
| 25 | 0.971 | 0.1543 |
| 26 | 0.969 | 0.1552 |
| 27 | 0.943 | 0.2466 |
| 28 | 0.969 | 0.1550 |
| 29 | 0.969 | 0.1564 |

 **Best 0.998 0.1288**

**2.6.2.4 Performance summary of feature selection**

The performance comparison of three feature selection algorithms - Correlation, Gain Ratio, and Information Gain - tested using sequential modeling from 4 to 30 features. The Information Gain algorithm produced the highest ROC value (0.998) and the lowest RMSE (0.1288) using just 17–18 features, according to the results, which are shown in Table 11. As a result, it is the only algorithm that has effectively achieved the study's feature selection goals by enhancing model accuracy and drastically lowering the number of features.

**Table 11: Performance summary of feature selection algorithms used for selecting the best features**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm**  | **Highest ROC value**  | **Lowest RMSE value**  | **Range of best features** |
| Correlation  | 0.993 | 0.1465 | 17-25 |
| Gain Ratio  | 0.997 | 0.1243 | 21-26 |
| Information Gain  | **0.998** | 0.1288 | **17-18** |

The results of feature selection show that using fewer features improves performance compared to using all of them. In comparison to all 30 features (ROC: 0.969, RMSE: 0.1564), the Information Gain algorithm performed best, obtaining optimal results with 17–18 features (ROC: 0.998, RMSE: 0.1288). Thus, this study eliminates the least significant 12 features and chooses the top 18 ranked features for the hybrid intrusion detection model.

The selected features are: src\_bytes, dst\_bytes, service, protocol\_type, srv\_count, flag, dst\_host\_same\_src\_port\_rate, dst\_host\_srv\_rerror\_rate, diff\_srv\_rate, dst\_host\_rerror\_rate, duration, dst\_host\_srv\_count, rerror\_rate, srv\_rerror\_rate, srv\_diff\_host\_rate, same\_srv\_rate, dst\_host\_diff\_srv\_rate, and dst\_host\_same\_srv\_rate.

* + 1. **Development of PKRIDS framework**

**2.6.3.1 Proposed PCAMIX-KPCA-RF Intrusion Detection System (PKRIDS)**

In order to handle unbalanced network data, this study proposes a hybrid intrusion detection system that combines machine learning and multivariate statistical control. Three elements are integrated into the PKRIDS framework: (1) A PCAMIX-Hotelling T2 chart that uses statistical thresholds and principal component analysis to identify anomalies; (2) KPCA-based control charts that use kernel transformations to identify nonlinear patterns; and (3) A Random Forest classifier that uses ensemble decision trees to categorize attacks. This cohesive method addresses the inherent class imbalance while efficiently detecting and classifying network anomalies by utilizing both statistical and machine learning techniques.

**Steps for PKRIDS Framework**

(a) **Hotelling's T2 Statistic for PCAMIX**:

Step 1: Compute the principal component scores $t\_{i}$ using the eigenvectors $V\_{i}$**​**and data points

$y\_{i}:t\_{i} $**​= ⟨**$V\_{i}$**​,** $Φ$ **(**$y\_{i}$**​) ⟩**

Step 2: Calculate the Hotelling's T2 statistic $(T^{2}\_{PCAMIX})$ using the principal component scores:
$T^{2}\_{PCAMIX}= \sum\_{i=1}^{m}λ\_{i}^{-1}t\_{i}t\_{i}^{T} $

where $λ\_{i}$​ denotes the eigenvalues associated with the principal components.

(b) **KPCA implementation for Data Transformation:**

Step 3: Transform the input data matrix $X$ using the KPCA algorithm:

Assuming that we have a dataset $X \in R^{m X n}$ where *m* is the number of observations and *n* is the number of features, KPCA maps the data into a higher-dimensional feature space **Φ(*X*)** using a kernel function

$k\left(x\_{i},x\_{j}\right):Φ:X\rightarrow Φ\left(X\right)\in R^{m X p}$

where ***p*** is the number of principal components to be retained.

Step 4: Obtaining the principal Component Scores:

The principal component scores *ZKPCA*​ are obtained by projecting the data onto the eigenvectors derived from the covariance matrix of the transformed data**:**

$Z\_{kpca}= Φ\left(X\right)V$**​**

where $V $represents the matrix of eigenvectors corresponding to the largest eigenvalues.

(c) **Incorporating Hotelling's** $T^{2}$**, KPCA into Random Forests:**

Step 5: Combine the original data features $X$, the Hotelling's $T^{2}$ statistic and KPCA into the Random Forest classifier: $Y=Random Forest ([X,T^{2}\_{PCAMIX},Z\_{kpca}])$

The hybrid intrusion detection model equation incorporating PCAMIX - Hotelling's $T^{2}$, KPCA, and Random Forests can be represented as follows in a single equation:

$Y\_{hybrid}=RF ([X,T^{2}\_{PCAMIX},Z\_{kpca}])$

where:

* $Y$ represents the predicted intrusion labels.
* $X$ is the original data matrix.
* $T^{2}\_{PCAMIX}$ denotes the Hotelling's $T^{2}$ statistic calculated based on the principal component scores obtained from *PCAMIX*.
* $Z\_{kpca}$ represents the transformed feature scores derived from the KPCA transformation.
* *RF* signifies the Random Forest classifier used to predict intrusion labels based on the combined feature set.

**2.6.3.2 Exploratory Visualization of Phase Two in the PKRIDS Framework**

Building upon the flowchart presented below, this section provides a detailed visual interpretation of Phase 2 using scatter plots and Hotelling’s T² control charts derived from PCAmix and KPCA components.

**Scatter plots of PCAMIX scores for the NSL\_KDD dataset, showing the separation of normal and attack instances across different dimensions (Dimensions 1-3, 4-6, and 7-9).**







Figure 2: Scatter plots of PCAMIX scores for the NSL\_KDD dataset

Different patterns across principal components are revealed by the scatter plot analysis. Despite some overlap, Dimensions 1-3 capture the most significant variance and exhibit the strongest class separation (red/blue clusters). Dimensions 4-6, which represent additional but less important information, show decreased discriminative power with increased overlap. Noise seems to dominate dimensions 7-9, which exhibit total class overlap with no discernible division.

Effective dimensionality reduction is demonstrated by the progression, where:

The primary discriminative structure is present in the first three dimensions, Minor additional variance is added by middle dimensions (4-6), The classification value of higher dimensions (7-9) is insignificant.

**Scatter plots of KPCA scores for the NSL\_KDD dataset, showing the separation of normal and attack instances across different dimensions (Dimensions 1-3, 4-6, 7-9, 10-12).**

Plot A: Dim 1-3



Plot B: Dim 4-6



**Plot C: Dim 7-9**



**Plot D: Dim10-12**



**Figure 3: Scatter plots of KPCA scores** **for the NSL\_KDD dataset**

The significance of these dimensions for dataset modeling is highlighted by Plot (A), which displays distinct red and blue clusters demonstrating distinct nonlinear relationships and class separation. Dense red clusters contrast with scattered blue points in the scatter plot (B) for dimensions 4-6, which shows partial separation and captures nonlinear variance but has less class differentiation than dimensions 1-3. Completely dispersed points with substantial overlap are seen in dimensions 7-9, suggesting that they mostly contain noise or very few nonlinear patterns. Plotting dimensions 10–12 shows a star-like pattern with centrally clustered blue points and outward-radiating red points, indicating that these dimensions capture distinct but ultimately unimportant dataset characteristics for intrusion detection.

**Combine features of Pcamix\_score and Kpca\_score for the NSL\_KDD dataset**

Plot A: Dim 1-3



Plot C: Dim 7-9



Plot E: 13-15



Plot B: Dim 4-6



Plot D: Dim 10-12



Plot F: Dim 16-18

Plot G: Dim 19-21



**Figure 4: Combine features of Pcamix\_scoreNSL and Kpca\_scoreNSL Using NSL\_KDD Dataset**

The graphs display the combined PCAmix-KPCA results, which improve outlier identification by utilizing KPCA's nonlinear pattern detection and PCAmix's mixed-data handling.

Strong clustering is evident in Dimensions 1-3, which also exhibit effective linear-nonlinear complementarity with distinct normal (blue)/attack (red) separation and discernible outliers. Partial separation with linear attack patterns versus scattered normal points is seen in dimensions 4-6. Heavy overlap is seen in dimensions 7-9, which mainly capture noise.

While dimensions 13–15 show dense normal clusters with linear attack spreads, dimensions 10–12 form non-discriminative star patterns. This pattern is maintained with weaker separation in dimensions 16–18. Complete overlap and little useful information are visible in dimensions 19–21.

The hybrid approach offers thorough anomaly identification across all dimensions by fusing the linear analysis of PCAmix with the nonlinear detection of KPCA.

**Control chart for combine features of PCAmix and KPCA Using NSL\_KDD Dataset**



**Figure 5: Control chart for combine features of PCAmix and KPCA Using NSL\_KDD Dataset**

Using a UCL of 17.8934, the chart applies Hotelling T2 to NSL-KDD (21 variables). 2,941 anomalies (19.4%) were identified as red points above UCL in an analysis of 15,191 observations.

Dense clusters (T2>800) in the first 5,000 observations suggest focused attacks. Sporadic anomalies in later observations point to changing threats. Baseline points are normal traffic.

Although the 21-feature method increases sensitivity, it requires verification to lower false positives. The pattern shows: Waves of early attacks, Subsequent isolated incidents, Possible fresh dangers. While careful pattern analysis is necessary to predict intrusions, this shows improved anomaly detection.

**Scatter plots of PCAMIX scores for the TON\_IoT dataset, showing the separation of normal and attack instances across different dimensions (Dimensions 1-3, 4-6, and 7-9).**

Plot A: Dim 1-3



Plot B: Dim 4-6



**Plot C: Dim 7-9**



**Figure 6: Scatter plots of PCAMIX scores for the TON\_IoT dataset**

Plot (A) for dimensions 1–3 shows clear red and blue point groups. The blue points disperse but line up in specific patterns; the red points bunch more together. These aspects reveal important contrasts. There is a lot of mixing in dimensions 4–6 where blue points are scattered and red points create little clusters. These aspects don't really help to define the classes. Plot C's scattered red and blue clusters imply that these dimensions pick up background noise or minor fluctuations.

**Scatter plots of KPCA scores for the TON\_IoT dataset, showing the separation of normal and attack instances across different dimensions (Dimensions 1-3, 4-6, 7-9,10-12 and 13-15).**

Plot A: Dim 1-3



**Plot C: Dim 7-9**



Plot B: Dim 4-6



**Plot D: Dim 10-12**



**Plot E: Dim 13-15**



**Figure 7: Scatter plots of KPCA scores for the TON\_IoT dataset**

Variations in class separation between normal and anomalous data points are exposed by scatter plot analysis over several KPCA dimensions. Dimensions 1–3 show the best separation, catching significant nonlinear relationships where normal points are more dispersed and anomalous points are tightly clustered, so indicating important variability in normal behavior. In Dimensions 4–6, the differences between classes lose their clarity even if some structural patterns still show. Plotting shows more overlap between normal and anomalous points as the study moves into Dimensions 7–9 and beyond, implying a drop in discriminative power. Dimensions 13–15 show dense overlap and little structure; Dimensions 10–12 show limited separation with starburst-like patterns. Early KPCA dimensions generally help greatly in anomaly detection; later dimensions mostly capture noise or low-variance patterns.

**Combine features for PCAmix and KPCA Using TON\_IoT Dataset**

**Plot A: Dim 1-3D**



**Plot C: 7-9D**



**Plot B: Dim 4-6D**



**Plot D: Dim 10-12D**



**Plot E: Dim 13-15D**



**Plot G: Dim 19-21D**



**Plot F: Dim 16-18D**



**Plot H: Dim22-25D**



**Figure 8: Scatter plots of Combine features for PCAmix and KPCA Using TON\_IoT Dataset**

Using PCAmix and KPCA features together on the TON\_IoT dataset shows different efficacy in different dimensions in separating normal from anomalous data. Though some overlap still, normal points cluster tightly in Dimensions 1–3 while anomalies are more scattered. Separations in Dimensions 4–6 get better; in Dimensions 7–9 and 10–12, where anomalies are further isolated and sometimes form subgroups, they get clearer. Dimensions 16–18 show modest separation; dimensions 19–21 and 22–25 show strong isolation of anomalies, so indicating their great value for outlier detection. These graphs show that whereas later dimensions capture more complicated patterns, different dimension sets help to identify anomalies in their own right. By using both linear and nonlinear feature, PCAmix and KPCA improve the performance of the hybrid IDS overall.

**T2 Control chart for the Combine features using TON\_Iot Dataset**



**Figure 9: T2 Control chart for the Combine features for PCAmix and KPCA Using TON\_IoT Dataset**

The Hotelling's T2 statistic for 12,102 samples utilizing 26 combined features from the TON\_IoT dataset is shown in this figure. Values above the upper control limit (UCL), which is displayed in red, were considered anomalies. The UCL was set at 27.1. Outliers were identified in 2,941 samples (about 19.4%). The first 5,000 samples show a dense concentration of anomalies that most likely correspond to particular attack scenarios. Significant departures from typical behavior were indicated by some T2 values that were higher than 800. Anomalies spread out more as the chart goes along, indicating changes in network behavior or novel intrusion patterns. The findings show that the detection of irregularities is improved by combining PCAmix and KPCA features. But doing so also makes the model more complex, necessitating thorough validation to reduce false positives and boost detection accuracy.

Flowchart of the proposed PKRIDS framework, showing the integration of PCAMIX, KPCA, and RF is illustrated in Figure 10.



Figure 10: Information flow of the proposed PKRIDS

This flowchart outlines the key components and implementation stages of the proposed PKRIDS (PCAmix - KPCA - Random Forest Intrusion Detection System). It illustrates how the system integrates statistical and machine learning techniques - beginning with data preprocessing and feature selection, followed by dimensionality reduction using PCAmix and KPCA, and concluding with anomaly detection and classification using control charts and a Random Forest model

3. results and discussion

The performance of the proposed model is evaluated using two distinct datasets: NSL\_KDD and TON\_IOT. The results are summarized in the confusion matrix presented in Table 12.

**3.1** **Performance evaluation of the PKRIDS**

PKRIDS demonstrated its efficacy in intrusion detection by achieving >99.7% detection accuracy on both datasets with remarkably low false alarm rates (<0.2%), as indicated in Table 12.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | Dimension | TP | FN | FP | TN | TP rate | FP rate | F1 | Hit (%) |
| NSL\_KDD | 10 | 10181 | 14 | 19 | 1180 | 0.9981 | 0.0018 | 0.9446093 | 99.71 |
|  |  |  |  |  |  |  |  |  |  |
| TON\_IOT | 12 | 8570 | 9 | 12 | 486 | 0.9986 | 0.0013 | 0.97199308 | 99.76 |

Table 12: Confusion matrix and other performance metrics

For the NSL\_KDD dataset, the model demonstrated strong performance with 10,181 true positives (TP), 14 false negatives (FN), 19 false positives (FP), and 1,180 true negatives (TN). It achieved a 99.81% true positive rate and an exceptionally low 0.18% false positive rate, with an F1 score of 0.9446 and overall hit rate of 99.71%.

The TON\_IoT dataset showed even better results, with 8,570 TP, 9 FN, 19 FP, and 486 TN. The model reached a 99.86% true positive rate, 0.13% false positive rate, 0.9720 F1 score, and 99.76% hit rate.

Additional evaluation using ROC curves, AUC, and P-R curves confirmed the model's effectiveness in handling imbalanced intrusion detection datasets, demonstrating reliable discrimination between normal and malicious traffic.



**Figure 11** : ROC Curve Vs AUC value for the NSL\_KDD dataset

With a TPR of 99.0% and an FPR of 0.78%, the ROC curve for NSL\_KDD (Fig 11) shows outstanding performance in our analysis. The model's strong ability to differentiate intrusions from regular traffic is confirmed by the curve's placement in the upper left corner. Only 79 real intrusions were overlooked, according to the high TPR, while the low FPR indicates few false alarms. With a high likelihood of accurately ranking positive instances above negative ones, the model's near-perfect classification ability is further validated by its remarkable AUC of 0.9975.

****

**Figure 12**: Precision-Recall (P-R) curve for the NSL\_KDD dataset

With a precision of 0.99, a recall of 0.98, and an AUC of 0.8816, the hybrid intrusion detection method (PKRIDS) is clearly very successful, as shown by the P-R curve for the NSL\_KDD dataset. This set of metrics demonstrates how well the model can identify intrusions while preserving a low false positive rate. The model's ability to distinguish between intrusions and non-intrusions across a range of thresholds is further supported by the AUC value. These findings collectively imply that the model offers a strong defence against possible threats and is appropriate for real-world network security applications.

****

**Figure 13: ROC Curve Vs AUC value for TON\_IOT**

The balance between True Positive Rate (TPR) and False Positive Rate (FPR) across classification thresholds is depicted by the ROC curve for the TON\_IOT dataset (Fig. 13). With a remarkable TPR of 99.65%, the model accurately detects almost all intrusions and misses only 40 real attacks. By infrequently misclassifying benign traffic as malicious, it maintains strong specificity with an FPR of 2.07%. The near-perfect AUC of 0.9975 validates the model's exceptional ability to distinguish between legitimate traffic and intrusions, despite the fact that this FPR is marginally higher than NSL-KDD's. These findings show strong performance with the ideal sensitivity-specificity ratio for real-world implementation.

****

**Figure 14: P-R Curve for TON\_IOT**

With precision=0.9989, recall=0.9965, and AUC=0.8819., the P-R curve analysis shows how effective PKRIDS is on the TON\_IoT dataset. The AUC shows consistent performance across thresholds, confirming the model's ability to detect intrusions accurately while minimizing false positives. According to the findings, PKRIDS is especially well-suited for protecting IoT environments from changing threats.

PKRIDS consistently strikes the ideal balance between low false positive rates and high true positive rates across the two assessed datasets. For operational network security systems where, alert fatigue must be prevented, this dual capability minimizes unnecessary alerts while ensuring dependable threat detection. These P-R curve results offer thorough confirmation of the robustness of PKRIDS when paired with ROC curve and confusion matrix results. The method's consistent high performance across all evaluation metrics makes it particularly valuable for enhancing security in IoT ecosystems where both detection accuracy and operational efficiency are paramount.

4. Conclusion

In order to efficiently identify anomalies in mixed-type, high-dimensional network data, this paper introduced PKRIDS, a hybrid intrusion detection model that combines PCAMIX, KPCA, and Random Forest. The model showed very low false positive rates and high detection accuracy when tested on the NSL-KDD and TON\_IoT datasets. The system's performance and interpretability were greatly improved by the application of feature integration and dimensionality reduction. The model's sensitivity to different attack patterns was further confirmed by exploratory visualizations and Hotelling's T2 control charts. Future research will concentrate on real-time deployment, dynamic threshold adaptation, and application in edge and IoT-based environments, even though the model works well in offline settings.

References

1. Ahsan, M., Mashuri, M., & Khusna, H. (2019). Intrusion detection system using bootstrap resampling approach of T² control chart based on successive difference covariance matrix. *Journal of Theoretical and Applied Information Technology*, 96(8).
2. Breiman, L. (2001). Random Forests. Machine Learning 45, 5–32. [https://doi.org/10.1023/A:1010933404324](https://doi.org/10.1023/A%3A1010933404324)
3. Kamini, C. N. (2020). Using Machine Learning and Statistical Models for Intrusion Detection. *International Journal of Computer Applications*, 175(31).
4. Mo, S., et al. (2021). PCA mix-based Hotelling's T² multivariate control charts for intrusion detection system. *IET Information Security*.
5. Axelsson S. (2000): *Intrusion Detection Systems: A Survey and Taxonomy. Technical Report 99-15*. Department of Computer Engineering, Chalmers University; 2000.
6. Kabiri P, Ghorbani AA. (2005): Research on intrusion detection and response: a survey. *Int J Netw Secur*. 2005;1(2):84-102. https://doi.org/10. 6633/IJNS.200509.1(2).05.
7. Ahmim A, Derdour M, Ferrag MA. (2018): An intrusion detection system based on combining probability predictions of a tree of classifiers. *Int J Commun Syst*. 2018;31(9): e3547. https://doi.org/10.1002/dac.3547.
8. Kamini C. Nalavade (2020): Using Machine Learning and Statistical Models for Intrusion Detection. International Journal of Computer Applications (0975 – 8887) Volume 175 – No. 31.
9. Rajkumar, Sheela & Rajendran, Vimala Devi & T, Subbu. (2015). *Assessment of Intrusion Detection System* - A Multi-Disciplinary Analysis.
10. Tang, C., Luktarhan, N., & Zhao, Y. (2020). SAAE-DNN: Deep Learning Method on Intrusion Detection. *Symmetry*, *12*(10), 1695. <https://doi.org/10.3390/sym12101695>
11. Schölkopf, B., Smola, A., Müller, K.-R., 1997. Kernel principal component analysis. In: Artificial Neural Networks—ICANN’97, vol.1, pp.583–588.
12. Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, pp.144–152.
13. E. Baker (2009), International Encyclopedia of Education (3rd edition), Oxford, UK: Elsevier, (In Press).
14. P. Mitra, C. A. Murthy and S. K. Pal. (2002): “Unsupervised feature selection using feature similarity,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 3, pp.301312, 2002.
15. Miller, A. (2002) Subset Selection in Regression. 2nd Edition, Chapman & Hall/CRC, New York. <http://dx.doi.org/10.1201/9781420035933>
16. H. Almuallim and T. G. Dietterich. (1994): “Learning boolean concepts in the presence of many irrelevant features,” Artificial Intelligence, vol. 69, no. 1-2, pp. 279–305, 1994.
17. Chavent, M., Kuentz-Simonet, V. & Saracco, J. (2012): Orthogonal rotation in PCAMIX. Adv Data Anal Classif 6, 131–146). <https://doi.org/10.1007/s11634-012-0105-3>
18. Chavent, M., Kuentz-Simonet, V., Labenne, A., & Saracco, J. (2014): Multivariate analysis of mixed data: The R package PCAmixdata. arXiv preprint arXiv:1411.4911.
19. Aneetha Avalappampatty Sivasamy, Bose Sundan (2015): A Dynamic Intrusion Detection System Based on Multivariate Hotelling’s T2 Statistics Approach for Network Environments, The Scientific World Journal, vol. 2015, Article ID 850153, 9 pages, 2015. <https://doi.org/10.1155/2015/850153>
20. Muhammad Ahsan, Muhammad Mashuri, Heri Kuswanto, Dedy Dwi Prastyo (2018). Intrusion detection system using multivariate control chart Hotelling’s T² based on PCA. Int J Adv Sci Eng Inf Technol, 8(5), ISSN: 2088-5334.
21. Muhammad Ahsan, Muhammad Mashuri, Hidayatul Khusna, Wibawati (2022). Kernel principal component analysis (PCA) control chart for monitoring mixed non-linear variable and attribute quality characteristics. Heliyon, <https://doi.org/10.1016/j.heliyon.e09590>
22. Aldallal, A. (2022). Toward: Efficient Intrusion Detection System Using Hybrid Deep Learning Approach. Symmetry, 14, 1916. https://doi.org/10.3390/sym14091916
23. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., Alazab, A. (2020). Hybrid Intrusion Detection System Based on the Stacking Ensemble of C5 Decision Tree Classifier and One Class Support Vector Machine. Electronics, 9(1), 173. <https://doi.org/10.3390/electronics9010173>
24. Moustafa, Nour, Slay, Jill (2017). A hybrid feature selection for network intrusion detection systems: Central points. <http://10.4225/75/57a84bd57fbefbb>