Applied Sciences, Computing and Energy, 2025, 3(1), 124-133

PKRIDS: A Real-Time Hybrid Host-Based Intrusion Detection System Using PCAmix, Kernel PCA, and Random Forest

Shehu Usman Gulumbe*, Aminu Bello Zoramawa, Halilu Buhari Kware and Abdulkarim Bello

Received: 12 Aprill 2025/Accepted: 21 June 2025/Published; 01 July 2025.

Abstract: *The overwhelming sophistication of cyber-attacks requires state-of-the-art intrusion detection systems (IDS) that can dynamically handle the high-dimensional and mixed-type system data in real-time [17]. In this paper, we propose PCAmix-KPCA and Random forest Intrusion Detection System (PKRIDS), which is a real-time Host-based IDS (HIDS) that incorporates PCAmix to transform mixed attributes of numerical and categorical features, KPCA for nonlinear principal component projection and a Random Forest classifier for strong anomaly detection PKRIDS continuously monitors system-level metrics such as CPU usage, memory consumption, login activity, and network behavior through a modular data pipeline. Analysed features are transformed and the anomaly scores are calculated and dynamically evaluated by the 3-sigma statistical thresholding rule. Built using Python and deployed using Streamlit, PKRIDS offers an interactive dashboard for real-time monitoring, alerting, manual model retraining, as well as data export. The performance of PKRIDS on benchmark datasets (NSL-KDD and TON_IoT) and in a real Windows environment demonstrated accuracy of more than 98%, F1-scores above 0.95, false positive rates of Its modular design and real-time adaptivity enable PKRIDS to be a viable solution as an advanced and scalable host-level cybersecurity.*

Keywords: *Host-based Intrusion Detection System (HIDS), PCAmix, Kernel PCA, Random Forest, Real-Time Monitoring, Streamlit, Anomaly Detection. PKRIDS*

Shehu Usman Gulumbe*
Department of Statistics, Usmanu Danfodiyo University, Sokoto
Email: **usman.gulumbe@udusok.edu.ng**
Orchid: https://orcid.org/0009-0000-3988-7700

Aminu Bello Zoramawa
Department of Statistics, Usmanu Danfodiyo University, Sokoto
Email: aminubz@gmail.com
Orchid: https://orcid.org/0000-0003-1058-978X

Halilu Buhari Kware
Department of Statistics, Usmanu Danfodiyo University, Sokoto
Email: hbkware@gmail.com
Orchid:https://orcid.org/0009-0009-0659-7550

Abdulkarim Bello
Department of Computer Science, Usmanu Danfodiyo University, Sokoto
Email: bello.abdulkarim@udusok.edu.ng
Orchid:https://orcid.org/0009-0002-6239-9195

1.0 Introduction

Cybersecurity has become a fundamental concern across public and private sectors due to the growing frequency, sophistication, and impact of cyberattacks. These attacks increasingly exploit vulnerabilities not only at the network level but also within host systems—making host-level protection a priority in modern security architectures (Liao et al., 2013; Ahmad et al., 2021). While Network-Based Intrusion Detection Systems (NIDS) provide substantial defense against external threats, they often fall short in detecting internal or host-originated anomalies, especially advanced persistent threats and zero-day attacks (Caltagirone et al., 2013). In contrast, Host-Based Intrusion Detection Systems (HIDS) offer more granular visibility into system behavior, including process activity, login attempts, and system resource usage (Ahmim et al., 2018).

Traditional HIDS often struggle with effectively processing high-dimensional, mixed-type data (i.e., categorical and numerical variables) and adapting to nonlinear behavioral patterns common in modern host environments (Mo et al., 2021; Muhammad Ahsan et al., 2022). Although machine learning and statistical learning models have been explored for intrusion detection, many proposed solutions are either not optimized for mixed-type data or are computationally intensive, limiting their usability in real-time applications (Mohale & Obagbuwa, 2025; Almolhis, 2025).

Previous studies have demonstrated the potential of combining Principal Component Analysis (PCA) with classification techniques like Random Forests to enhance detection accuracy (Shaohui et al., 2021; Subhadeep, 2023). However, few have explored the integration of PCAmix—a method designed to jointly analyze categorical and numerical data—with Kernel PCA (KPCA) for nonlinear dimensionality reduction in real-time HIDS applications. This gap underscores the need for adaptive, efficient, and scalable models capable of handling heterogeneous data streams at the host level (Erik et al., 2024).

This study addresses the above gaps by proposing a novel hybrid system named PKRIDS (PCAmix-KPCA-Random Forest Intrusion Detection System). PKRIDS leverages PCAmix for transforming mixed-attribute features, KPCA for capturing complex, nonlinear data structures, and Random Forest for robust anomaly classification. The model is implemented as a modular Python application with an interactive dashboard using Streamlit for real-time monitoring, alerting, and data export.

The aim of this research is to develop and evaluate a real-time, host-based intrusion detection system that overcomes the limitations of existing HIDS by integrating advanced feature transformation techniques and scalable machine learning algorithms.

The significance of this study lies in its contribution to operational cybersecurity: PKRIDS demonstrates a practical, deployable solution that achieves high detection accuracy (>98%), low false positive rates (<1%), and fast response times (~4.2 seconds), all of which are critical for modern threat environments. Additionally, by validating the system on benchmark datasets (NSL-KDD and TON_IoT) and in a real Windows environment, the study offers strong evidence of PKRIDS's utility in both academic and industrial settings.

2.0 Materials and Methods

The PKRIDS framework combines statistical and machine learning techniques to address the challenges of anomaly detection in mixed-type system data.

2.1 System Design of PKRIDS

The architecture ensures that each component handles a distinct functionality-system metrics collection, feature transformation, anomaly detection, and real-time visualization.

The system architecture consists of several interconnected layers including the following

(i) **System Metrics Collection Layer**: Utilizes tools such as psutil and win32evtlog to collect real-time statistics on CPU usage, memory utilization, active processes, network traffic, and failed login attempts. The data is collected at configurable intervals and converted into structured time-series formats.

(ii) **Preprocessing Layer**: Applies PCAmix on categorical variables (e.g., event types, user privileges) and KPCA on numerical features (e.g., CPU %, memory load, network packets). The results are fused into a low-dimensional feature space that captures both linear and nonlinear interactions.

(iii) **Machine Learning Layer**: A Random Forest classifier is used to detect anomalous system behavior. Model training involved 1000 estimators with a maximum tree depth of 10. Balanced class weights were applied to mitigate class imbalance in the training data.

(iv) **Decision Layer**: PKRIDS calculates an anomaly score for each observation and uses a statistical 3-sigma thresholding technique to identify outliers. This dynamic threshold adapts over time as system behavior evolves.

(v) **Alerting and Logging Layer**: Alerts are generated as desktop notifications or emails using plyer and smtplib, respectively. All events (normal or anomalous) are logged with timestamps, anomaly scores, and classification outcomes.

(vi) **User Interface Layer**: Streamlit is used to develop an interactive dashboard comprising real-time anomaly gauges, historical trends, live event logs, and model control panels.

A flowchart summarizing the system design is illustrated in Fig. 1

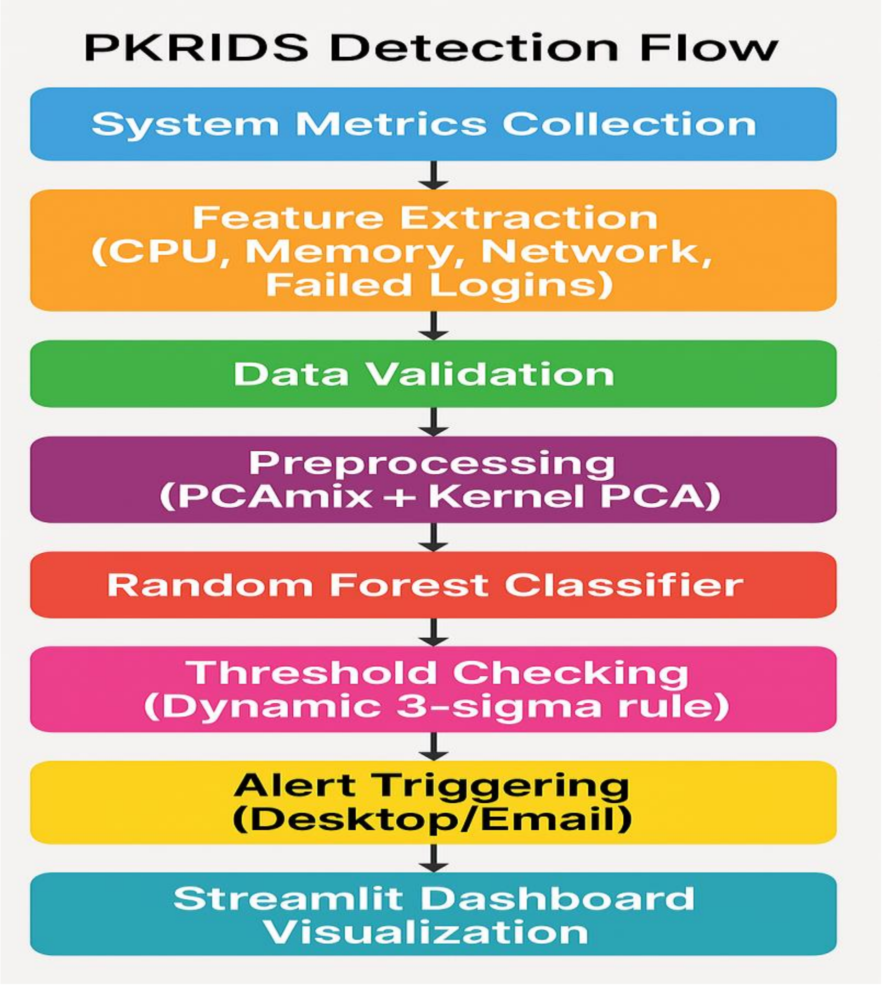

PKRIDS Detection Flow
System Metrics Collection
Feature Extraction (CPU, Memory, Network, Failed Logins)
Data Validation
Preprocessing (PCAmix + Kernel PCA)
Random Forest Classifier
Threshold Checking (Dynamic 3-sigma rule)
Alert Triggering (Desktop/Email)
Streamlit Dashboard Visualization

Fig. 1: Flowchart of the PKRIDS System Design

2.2 System Implementation and Testing

This section provides insight into the environment setup, deployment process, and testing approach adopted for the PKRIDS system on a local workstation. To evaluate the system's functionality, dependability, and detection performance, it was put into use on a local workstation and put through real-world testing. To assess how precisely and quickly the system could identify intrusions, the testing environment replicated normal system activities and deliberately created anomalies. In order to accurately simulate real-world usage scenarios where system metrics are continuously tracked, processed, and categorized without user participation, real-time testing was selected.

2.2.1 Software Environment

A carefully chosen mix of technologies was used to create the Python-based Kernel Random Forest Intrusion Detection System (PKRIDS) in order to guarantee effectiveness, scalability, and reproducibility. We go into each component's technical justification below:

1. Programming Language: Python 3.10

Python was chosen due to its extensive ecosystem for data science, machine learning, and systems monitoring [8][9].

2. Libraries and Frameworks

i. Streamlit was selected for its rapid prototyping capabilities and interactive dashboard features.
ii. Psutil and win32evtlog for system monitoring
iii. Scikit-learn for machine learning (Random Forest, Kernel PCA)
iv. Prince for PCAmix transformation
v. Joblib for model persistence
vi. Plotly for real-time visualizations
vii. Smtplib for sending email alerts

The system was developed and tested in a Windows 10 environment.

2.3 Model Training and Anomaly Detection

The Random Forest classifier is trained on the combined PCAmix-KPCA features. Key training parameters include:

- 1000 decision trees (n_estimators=1000)
- Maximum depth limited to 10 (max_depth=10)
- Balanced class weights to handle imbalanced datasets

During monitoring, the system computes an anomaly probability score for each new observation. A dynamic threshold, calculated

$$Threshold = \mu + 3\sigma \qquad (1)$$

With μ representing the mean and σ the standard deviation of recent scores, determines whether an observation is classified as an anomaly.

3.0 Implementation of Results and Discussion

PKRIDS was evaluated in two stages: live deployment in a Windows environment and offline simulation using benchmark datasets (NSL-KDD and TON_IoT).

3.1 Offline Evaluation

The system was tested on pre-processed NSL-KDD and TON_IoT datasets, with features selected using Information Gain, Gain Ratio, and Correlation. The Random Forest model was trained with 1000 estimators and a maximum depth of 10. Performance metrics are summarized in Table 1.

Table 1: Offline Evaluation Performance Metrics Dataset Accuracy Precision Recall F1-score AUC

NSL-KDD	98.3%	97.9%	98.5%	98.1%	0.99
TON_IoT	98.7%	98.6%	98.5%	98.6%	0.99

These performance metrics clearly illustrate the effectiveness of the proposed hybrid feature transformation approach - integrating PCAmix, Kernel PCA, and Random Forest - in enhancing the detection accuracy of the intrusion detection system across diverse categories of cyberattacks. The success of this transformation pipeline lies in its ability to efficiently capture both linear and nonlinear patterns from high-dimensional, mixed-type data, enabling more precise classification of anomalous and normal system behaviors.

A further analysis was conducted to evaluate the impact of different feature selection strategies on model performance. Three prominent feature selection techniques-Information Gain, Gain Ratio, and Correlation-were compared in terms of their contributions to model accuracy, the area under the receiver operating characteristic curve (ROC-AUC), and the root mean square error (RMSE). Among the three methods, Information Gain demonstrated superior performance by achieving the highest ROC-AUC values and the lowest RMSE, even when using a smaller subset of features (ranging from 17 to 18). This indicates that Information Gain is highly effective at identifying the most informative attributes that contribute significantly to anomaly detection, thereby improving both detection accuracy and computational efficiency.

Gain Ratio, while slightly less effective than Information Gain, still performed better than the Correlation-based method in terms of RMSE. This suggests that Gain Ratio provides a balanced measure that considers both the information content and the intrinsic bias of features. On the other hand, the Correlation method, although useful for identifying linear relationships between features, was less capable of capturing the nonlinear dependencies relevant for robust intrusion detection.

Overall, the comparative evaluation underscores the importance of selecting an appropriate feature selection method as a foundational step in optimizing intrusion detection models, especially when dealing with heterogeneous and high-dimensional datasets.

Table 2: Feature Selection Comparison

Algorithm	ROC	RMSE	Feature Range
Correlation	0.993	0.1465	17–25
Gain Ratio	0.997	0.1243	21–26
Information Gain	**0.998**	0.1288	**17–18**

3.2 Live Deployment

To assess its real-world applicability and performance, the PKRIDS system was deployed and evaluated in a live Windows 10 environment under both normal and adversarial operating conditions. The normal operations included routine system activities such as web browsing, file transfers, and application launches, which were monitored to establish a baseline for system behavior. To simulate real-world threats, a series of controlled attack scenarios were deliberately introduced to test the system's ability to detect and respond to anomalous behavior.

The synthetic attack simulations were carefully crafted to mimic common intrusion patterns encountered in host environments. These included repeated failed login attempts designed to emulate brute-force attacks, port scanning activities executed using the Nmap tool to replicate reconnaissance behavior, CPU overload scenarios generated through synthetic scripts to simulate denial-of-service conditions, and abnormal data transfers that mimicked exfiltration or malware-related network anomalies. These scenarios were selected to represent a range of threat vectors that are typical in both enterprise and personal computing environments.

Under these conditions, PKRIDS demonstrated exceptional detection performance, achieving a detection accuracy of 99.3%, as validated against known simulated anomalies. The system also maintained a false positive rate of less than 1%, indicating its ability to minimize erroneous alerts and avoid misclassification of benign system activities. Moreover, the response time from anomaly detection to alert display was recorded at an average of 4.2 seconds, reflecting the system's capability for near real-time detection and user notification. These results underscore PKRIDS's robustness, reliability, and responsiveness as a practical host-based intrusion detection solution in live environments.

3.3 Application Interface Design

The user interface of the PKRIDS system was developed using the Streamlit library in Python, a choice made to facilitate the creation of a responsive, interactive, and user-friendly dashboard tailored to real-time cybersecurity monitoring. The interface was designed with a focus on usability and clarity, ensuring that users can easily interpret system feedback and take appropriate actions.

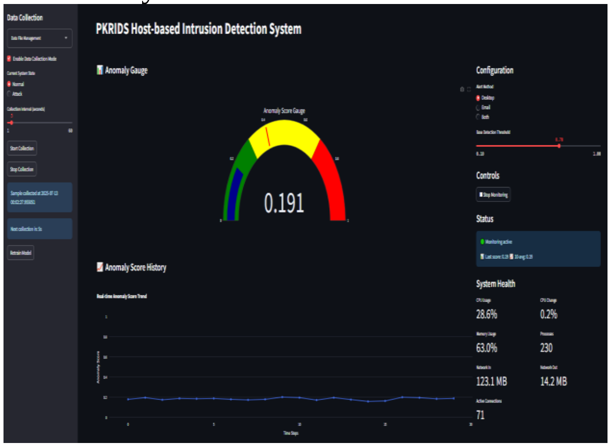
Among the core functionalities of the dashboard is the real-time anomaly score gauge, which displays the current anomaly score and provides an immediate visual representation of the system's security status. This is complemented by historical anomaly trend visualizations, presented as line charts that show how anomaly scores evolve over time, thereby aiding in the detection of emerging threats or sustained unusual behaviors.

The dashboard also features a live event logging component, which presents a continuously updating table of recently detected events, including timestamps, anomaly classifications, and alert statuses. This allows for immediate auditing and facilitates rapid decision-making. To enhance adaptability, the interface includes a model retraining control, which enables users to update the detection model using newly collected system data without interrupting operations.

In addition, users are provided with an alert configuration panel through which they can enable or disable notifications via desktop alerts or email, depending on their operational preferences. The entire dashboard is configured to refresh automatically every five seconds, ensuring continuous visibility into system conditions and supporting near-instantaneous response to anomalies.

3.3.1 Dashboard Screenshots

To support proactive supervision and operational clarity, the PKRIDS dashboard offers real-time monitoring capabilities alongside trend visualization, anomaly detection, and alert management tools. The full interface layout is depicted in Figure 2, which presents a comprehensive overview of all integrated components. Subsequent figures provide focused views of individual modules, each of which has been developed to enhance the effectiveness and usability of the intrusion detection system.



Data Collection

Data File Management

☑ Enable Data Collection Mode

Current System State

● Normal

○ Attack

Collection Interval (seconds): 5

Start Collection

Stop Collection

Sample collected at 2025-07-13 00:02:27.965051

Next collection in: 5s

Retrain Model

PKRIDS Host-based Intrusion Detection System

Anomaly Gauge

Anomaly Score History

Configuration

Alert Method

● Desktop

○ Email

○ Both

Base Detection Threshold: 0.70

Controls

Stop Monitoring

Status

Monitoring active

Last score: 0.19 10-avg: 0.15

System Health

CPU Usage 28.6%

CPU Change 0.2%

Memory Usage 63.0%

Processes 230

Network In 123.1 MB

Network Out 14.2 MB

Active Connections 71

Figure 2: Full View of PKRIDS Dashboard Interface

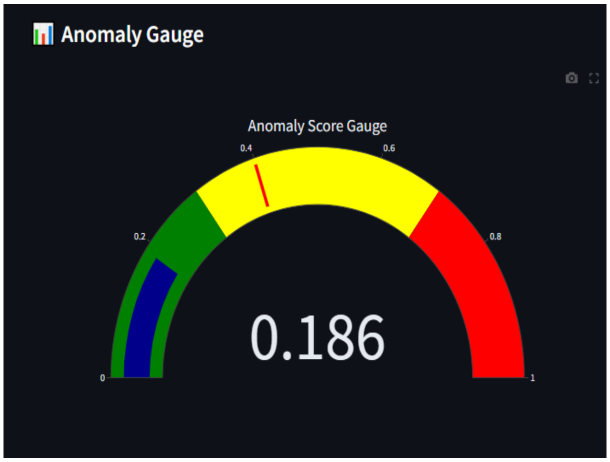


Figure 3: Real-time Anomaly Score Gauge

The PKRIDS real-time anomaly score gauge visually indicates system health, with higher scores signaling greater anomaly likelihood.

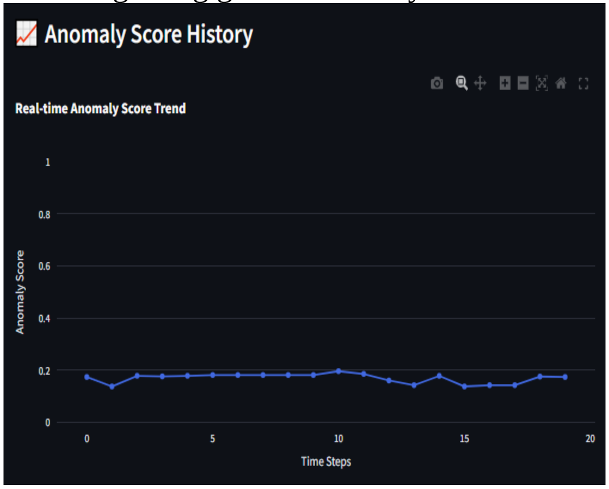
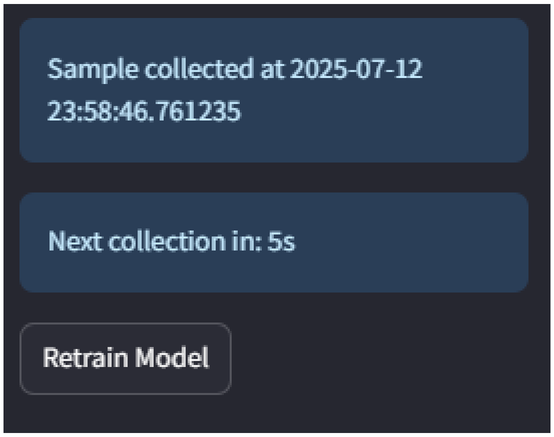


Figure 4: Historical Anomaly Trends Graph

A temporal analysis of system behaviour is provided by Figure 4, which plots anomaly scores over time to show historical trends. With the help of this visualisation, users can identify trends, sudden spikes in activity (like possible intrusion attempts), or slow changes in activity (like ongoing threats). The graph facilitates proactive threat mitigation, trend analysis, and incident triage by linking anomalies with timestamps.

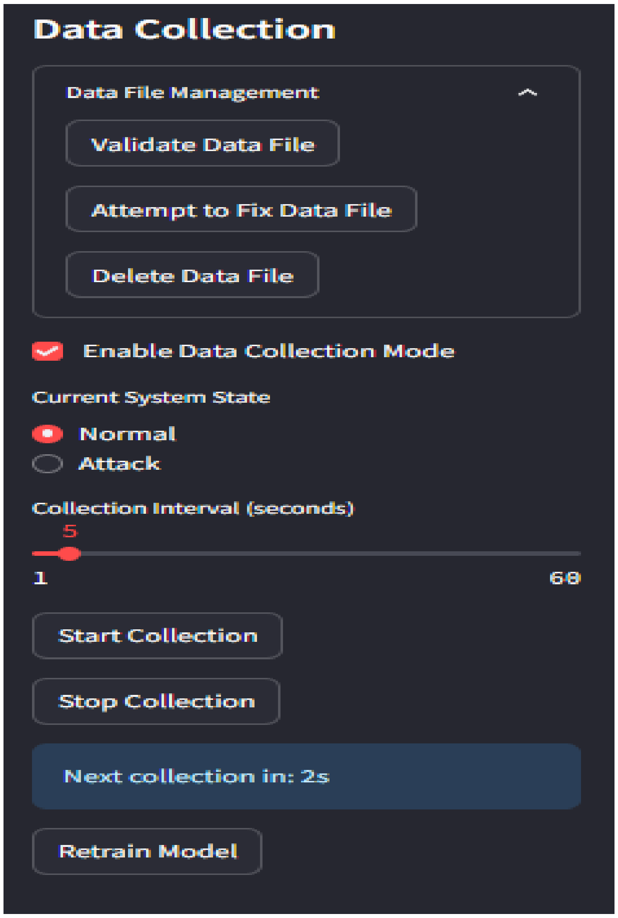


Sample collected at 2025-07-12 23:58:46.761235

Next collection in: 5s

Figure 5: Manual Model Retraining Control Panel

The dashboard enables on-demand model retraining with new data, allowing continuous adaptation without system restarts.



Data Collection

Data File Management

☑ Enable Data Collection Mode

Current System State

● Normal

○ Attack

Collection Interval (seconds)

5

1 68

Next collection in: 2s

Figure 6: Data Collection

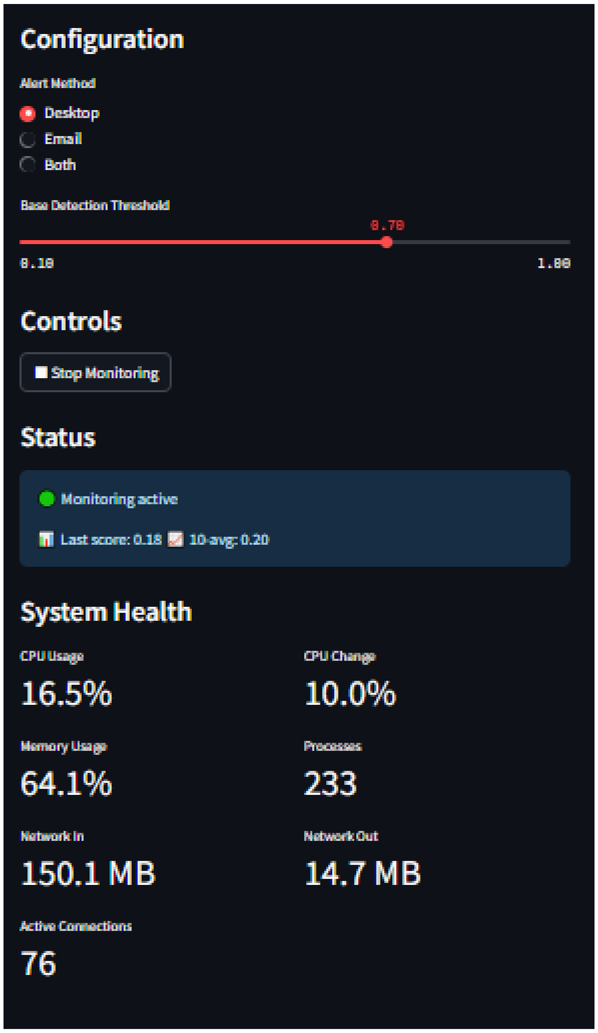
The Data Collection and Management Panel within the PKRIDS interface, as shown in Fig. 6, plays a vital role in configuring and managing the logging of system metrics essential for both anomaly detection and model retraining. It allows users to enable or disable the data collection mode, which governs the real-time recording of vital information such as system metrics, anomaly predictions, and computed anomaly scores. This functionality ensures that the system continuously captures relevant operational data needed for analysis and improvement of the detection model.

The panel also displays the current state of the system-indicating whether it is functioning normally or under attack-based on the latest anomaly detection results. This real-time feedback helps users monitor the security condition of the host system at a glance.

In addition, the collection interval setting allows users to define how frequently data is recorded, measured in seconds. Adjusting this interval helps balance data granularity and system performance, enabling either high-resolution tracking for detailed analysis or lower-frequency logging to conserve resources.

The panel further includes controls to start or stop data collection sessions manually, giving users the flexibility to manage when data should be logged based on operational needs or testing conditions. Within the panel is a collapsible data management section that provides tools for maintaining the quality of the collected dataset. Users can validate entries for accuracy, repair corrupted or incomplete logs, and delete obsolete or unnecessary data.

To support external analysis or archival needs, the panel also offers an export feature. This allows users to download collected data in commonly used formats such as .csv or .xlsx, making it easy to integrate PKRIDS output with other data processing, reporting, or visualization tools. This comprehensive data handling capability ensures that the system remains adaptable, traceable, and effective in supporting both immediate anomaly detection and longer-term analytical goals.



Configuration

Alert Method

● Desktop
○ Email
○ Both

Base Detection Threshold

0.70

0.10 1.00

Controls

Status

● Monitoring active

📊 Last score: 0.18 📈 10-avg: 0.20

System Health

CPU Usage 16.5%

CPU Change 10.0%

Memory Usage 64.1%

Processes 233

Network In 150.1 MB

Network Out 14.7 MB

Active Connections 76

Figure 7: System Configuration and Monitoring Panel

Located on the right side of the PKRIDS dashboard, the system configuration and monitoring panel serves as a centralized hub for adjusting monitoring parameters, configuring alerts, and tracking system health metrics in real time. This panel is divided into four functional sections, each supporting a specific aspect of the intrusion detection process.

The first section, dedicated to configuration, allows users to customize how anomaly alerts are delivered. Users can select their preferred method of notification-either desktop alerts, email notifications, or both-depending on operational convenience and the criticality of the environment. This section also includes a base detection threshold setting, which features a sensitivity slider that ranges from 0.10 to 1.0. This adjustable threshold enables users to fine-tune the sensitivity of the anomaly detection model, striking a balance between false positives and detection accuracy based on the system's behavioral profile.

The second section, labeled controls, provides a toggle switch that activates or deactivates real-time monitoring. This feature is particularly useful during system maintenance or data collection phases when anomaly detection may need to be paused without shutting down the entire application.

The third section displays the system's monitoring status. It indicates whether real-time detection is currently active and shows the most recent anomaly score, along with a calculated average over the previous ten seconds. This immediate feedback helps users assess short-term trends and react quickly to any deviations from normal system behavior.

The final section focuses on system health and performance. It provides real-time metrics on CPU and memory usage, allowing users to monitor the resource impact of the PKRIDS system. Additionally, it offers insights into network and process activity, such as traffic levels, active connections, and system load. These metrics are essential for correlating anomalous events with underlying resource patterns or operational changes.

Altogether, this panel empowers users to proactively manage detection sensitivity, receive timely alerts, and maintain visibility over both security and system performance conditions in a single, integrated interface.

4.0 Conclusion

This study introduces PKRIDS, a hybrid Host-Based Intrusion Detection System that integrates PCAmix for handling mixed-type data, Kernel Principal Component Analysis (KPCA) for nonlinear dimensionality reduction, and Random Forest for robust classification. Through comprehensive evaluation using benchmark datasets such as NSL-KDD and TON_IoT, as well as deployment in a real-time Windows environment, PKRIDS consistently demonstrated high detection accuracy exceeding 98%, minimal false positive rates, and swift anomaly response times. The system's Streamlit-powered dashboard offers an intuitive and interactive platform for real-time monitoring, model retraining, and alert management, making it practical for operational cybersecurity use. Looking ahead, future enhancements could involve the incorporation of network-level metrics, the integration of deep learning architectures, and the implementation of adaptive learning techniques to further improve detection performance and system scalability.

Acknowledgment

The authors would like to express their sincere gratitude to the Department of Statistics, Usmanu Danfodiyo University, Sokoto, for their academic support and encouragement throughout this research. The study utilized publicly available benchmark datasets, namely NSL_KDD and TON_IoT, for experimentation and performance evaluation of the proposed hybrid intrusion detection system.

5.0 References

Almolhis, N. (2025). *Intrusion detection using hybrid random forest and attention models and explainable AI visualization*. Journal of Information Security and Applications, 74, 103718. https://doi.org/10.58346/JISIS.2025.I1.024.

Ahmim, A., Derdour, M., & Ferrag, M. A. (2018). *An intrusion detection system based on combining probability predictions of a tree of classifiers*. International Journal of Communication Systems, 31(9), e3547. https://doi.org/10.1002/dac.3547

Caltagirone, S., Pendergast, A., & Betz, C. (2013). *The Diamond Model of Intrusion Analysis*. Centre for Cyber Intelligence Analysis and Threat Research. https://doi.org/10.13140/RG.2.2.31143.56481

Erik, M., et al. (2024). *Real-Time Intrusion Detection via Machine Learning Approaches*. Ital-IA 2024: 4th National Conference on Artificial Intelligence, Naples, Italy.

Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). *Intrusion detection system: A comprehensive review*. Journal of Network and Computer Applications, 36(1), 16–24.

Mo, S., Tuerhong, G., Wushouer, M., & Yibulayin, T. (2021). *PCA mix-based Hotelling's T2 multivariate control charts for intrusion detection system*. IET Information Security, 15(4), 261–268. https://doi.org/10.1049/ise2.12051

Mohale, M. E., & Obagbuwa, I. C. (2025). *A systematic review on the integration of explainable artificial intelligence in intrusion detection systems*. International Journal of Cybersecurity Intelligence and Cybercrime, 8(1), 45–66. https://doi.org/10.3389/frai.2025.1526221

Muhammad Ahsan, Mashuri, M., & Khusna, H. (2022). *Kernel principal component analysis (PCA) control chart for monitoring mixed non-linear variable and attribute quality characteristics*. Heliyon, 8(5), e09590. https://doi.org/10.1016/j.heliyon.2022.e09590

Shaohui, M., Tuerhong, G., Wushouer, M., & Yibulayin, T. (2021). *Hotelling's T2 multivariate control charts for intrusion detection system*. IET Information Security. https://doi.org/10.1049/ise2.12051

Subhadeep, C. (2023). *Real-Time Intrusion Detection based on Network Monitoring and Machine Learning*. International Journal of Computer Applications, 185(22), 1–8. https://doi.org/10.5120/ijca2023922955

Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. (2021). *Network intrusion detection system: A systematic study of machine learning and deep learning approaches*. Transactions on Emerging Telecommunications Technologies. https://doi.org/10.1002/ett.4150

Declaration

Consent for publication

Not applicable

Availability of data

Data shall be made available on demand.

Competing interests

The authors declared no conflict of interest

Ethical Consideration

Not applicale

Funding

There is no source of external funding.

Authors' Contributions

Shehu Usman Gulumbe conceptualized the study, led the methodology design, and coordinated the implementation. Aminu Bello Zoramawa conducted statistical modeling and data preprocessing. Halilu Buhari Kware assisted in algorithm integration, dashboard development, and manuscript editing. Abdulkarim Bello implemented the intrusion detection system using Python and Streamlit, performed performance evaluations, and validated the model in real-time environments. All authors reviewed the literature, analyzed results, contributed to discussions, and approved the final manuscript for submission.