Journal of Intelligent System and Applied Data Science (JISADS)

Journal homepage : https://www.jisads.com
ISSN (2974-9840) Online

INTELLIGENT HYBRID INTRUSION DETECTION SYSTEMS FOR IOT-CLOUD ENVIRONMENTS

Chahira Lhioui

Department of Computer Science and Artificial Intelligence, College of Computing and Information Technology, University of Bisha, Saudi Arabia
shahira@ub.edu.sa

ABSTRACT

The development of innovative mobile technologies with intelligent processes have led to the large-scale interconnection of heterogeneous devices and systems via the Internet of Things (IoT). Thus, cloud technology is considered the basis for IoT. Yet, in reality, numerous security challenges face these network advancements in the form of network anomalies and threats. In fact, IoT-Cloud networks are vulnerable to attacks due to their wireless communication features and the participation of IoT devices. Thus, security has become an important issue for the basic functionality of such networks. Ensuring the reliability of collected data remains a challenging issue. IoT devices especially are too vulnerable to attackers due to their limited security resources. In this context, traditional intrusion detection systems (IDS) are becoming incompatible with the new network environment, while systems related to machine learning and deep learning are emerging. Artificial Intelligence (AI) tools are powerful techniques that could be used to achieve this purpose. In this work, we propose applying a machine-learning technique with a real IoT dataset to develop an intelligent intrusion detection system (IIDS). Data transferred between interconnected systems in the IoT-cloud networks will be encrypted. Moreover, we propose an optimal intelligent solution for the IoT energy consumption problem. This solution is based also on a machine learning tool. Our proposed solution demonstrates 99% higher scalability than existing attack detection schemes and yields significant improvements in both security system performance and quality of service in IoT–cloud environments.

Keywords: IoT, Cloud, Security, QoS, Intrusion detection, Artificial Intelligence, Energy, Classification techniques.

1. Introduction

The expansion of IoT networks and their applications in different domains such as healthcare and smart homes has provided several everyday advantages which have significantly impacted our lives. With its low cost and flexibility, cloud computing has recently evolved in many areas including commercial, health, military, and education applications. A feature that is common to all these paradigms is the issue of wireless communication [1]. Over time, the evolution of IoT-cloud systems and their vulnerability to their socio-economic environment has led to new security needs. In fact, users consistently rely on the service providers to ensure data security and resource availability [2].

These information systems, including database systems, are exposed to threats of any kind arising from malicious users. Submitted data is at risk of alteration and destruction. Thus, we can deduce that the security challenge facing IoT-cloud expansion is basically composed of two main threats [3][4]:

Hacking cloud servers via the internet represents a significant threat to IoT architectures. Such attacks can

Online: https://doi.org/10.64680/jisads.v3i2.59

result in the theft of sensitive data, which is the most common risk, or allow attackers to gain control over IoT devices for malicious purposes.

- Hacking the cloud server through the IoT sensors themselves is possible considering that some IoT sensors are installed in remote non-secured areas.

To address the security challenges inherent in IoT–cloud environments, this research proposes a scalable and intelligent attack detection solution aimed at enhancing both system security and the quality of service. The main objectives of this study are as follows: (i) to design an efficient detection framework capable of handling large-scale and heterogeneous IoT–cloud data, (ii) to improve detection accuracy and system scalability while minimizing computational and resource overhead, and (iii) to evaluate the proposed solution using standard performance metrics and benchmark it against recent attack detection approaches.

Recent work in this domain highlights multiple categories of benchmark techniques. Traditional signature-based intrusion detection systems provide efficient identification of known threats but struggle with adaptability and scalability in dynamic environments. Machine learning–based frameworks have been shown to enhance detection performance through automated feature learning and classification, yet they may incur significant training overhead and latency in large IoT deployments. Deep learning–based intrusion detection systems demonstrate strong capabilities for recognizing complex and evolving attack patterns, achieving high accuracy in cloud and IoT contexts. For example, recent deep learning-enabled IDS models achieve notable performance improvements for IoT security tasks, while explainable AI techniques have been integrated to improve model interpretability in intrusion detection applications. Meanwhile, hybrid and federated learning strategies have been explored to balance detection effectiveness with distributed resource constraints and privacy considerations. These benchmark approaches reflect the current state of research and provide a comprehensive basis for comparing the effectiveness and scalability of the proposed framework. Malicious attacks can disrupt network performance by dropping or altering transmitted messages or exposing confidential information contained within them. If an external attacker gains access to user privileges and credentials, they can intercept transactions, manipulate data, provide falsified information, and overwhelm system services and resources. Such attacks can damage an organization's reputation, reduce productivity, and potentially result in significant financial losses and a decline in client trust [5].

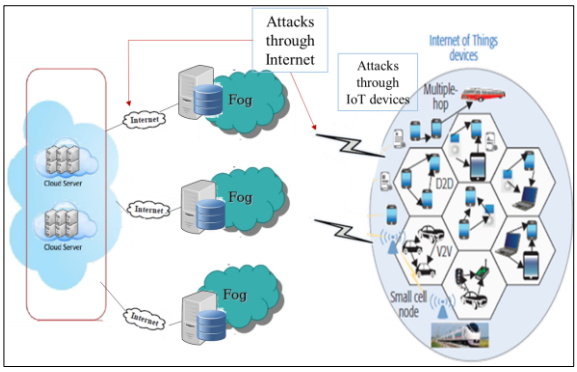

Figure 1. Overview of possible attacks facing IoT-Cloud

Furthermore, insider attacks are one of the most dangerous threats that face many systems today. An insider attack is carried out by people who are legitimately authorized in the system to perform certain tasks but decide to abuse this trust and harm the organization by causing breaches in the confidentiality, integrity, or availability of the organization's assets. Preventing insider attacks is a daunting task. Hence, it is necessary to develop a system capable of finding a middle ground where the necessary privileges are provided, and insider threats are mitigated.

The confidentiality, integrity, and availability of such data is achieved by designing a secure model such as an intrusion detection system (IDS). Although the proposed security schemes in the literature are numerous, there are still several important issues remaining in terms of security and performance in Cloud-IoT networks [6].

Therefore, among the well-used techniques of data security, we have the Intrusion Detection System (IDS), which is a type of detection system that protects networks by offering monitoring services for parts of our system. Moreover, an IDS is an essential component of a network's security, where intrusion and other security gaps must be discovered rapidly and effectively. However, IDS deals only with the security aspects and not energy efficiency. In this regard, another challenge facing the IoT expansion is the problem of energy consumption represented in the internet bandwidth consumption and the problem of cloud data storing [6].

In fact, energy efficiency has become a major concern in data centers and IoT sensors due to the environmental impact, cost, and operational expenses. The infrastructure requires resource planning in order to address the issue of energy waste, although this has recently been reduced substantially due to the utilization of best-practice technologies [7].

Online: https://doi.org/10.64680/jisads.v3i2.59

However, company managers are still dubious about the use of IoT as they fear problems related to security and energy consumption. Thus, they keep postponing the integration of IoT and then missing all the benefits provided by this new way of collecting data. In fact, because of the emergence of various breaches, several attacks are continuously occurring in IoT and cloud environment. In fact, a large portion of these threats is comprised of small variations of recently known cyberattacks. Accordingly, ensuring privacy and security to IoT users is one of the timeliest and urgent research issues.

In response to managers fears toward IoT integration, cloud suppliers have been trying to develop effective cloud services. Several experts around the world have proposed algorithm architecture and rules to make the cloud computing environment more secure and energy efficient.

The primary objective of machine learning systems is to facilitate seamless interaction between humans and computers. Due to their superior feature extraction capabilities, machine learning algorithms have delivered significant advantages across various security domains. Accordingly, our aim is to incorporate intelligent algorithms into the core functionalities of the intrusion detection system (IDS) to enhance data security while reducing energy consumption.

Consequently, a major challenge lies in developing a comprehensive model for secure data transfer and management, while ensuring optimal energy allocation across all system components. This challenge becomes even more critical given the performance demands of IoT–Cloud environments, which are constrained by low-latency access requirements. Therefore, the objective is to design a robust and secure model that simultaneously ensures high service performance and strong data protection.

In this paper, we address all the above critical issues and propose a secure scheme with an intelligent IDS to further enhance the data security and privacy of cloud/IoT networks. Moreover, we design an energy efficient system to improve the performance of these environments.

Therefore, the main objectives of this research are as follows:

1. To design an intelligent intrusion detection system for IoT–cloud environments that enhances data security and user privacy.
2. To integrate machine learning techniques into the IDS to improve detection accuracy and adaptability against both external and insider attacks.
3. To develop an energy-efficient framework that optimizes resource utilization and reduces energy consumption without degrading system performance.
4. To evaluate the proposed model against benchmark approaches in terms of security effectiveness, energy efficiency, and overall performance.

The rest of the paper is divided as following. Section 3 presents some related works about security and energy challenges in the IoT-cloud environments. Meaning while, we will present our proposed intelligent hybrid IDS. In the fourth section, we evaluate the performance of our proposal. Finally, we conclude our manuscript by conclusion and future work in section 5 and 6.

3. Literature Review

3.1 The Security Challenge

The Internet of Things (IoT) has a significant security problem that may have a strong impact on its expansion and effect on people's lives, business operations, and government services. In fact, hackers are always in the search for minor security flaws, and they are often successful in their threats to steal identity and financial data, etc. The user can no longer be confident in his anonymity or privacy.

Traditional security methods are no longer sufficient; businesses must evaluate a wide and diverse number of threats and investigate a more active solution to repair network breaches, detect risks, and restore system functioning. Because all types of data are now available on the internet, these attackers are constituted of professional technicians. Moreover, they have high-tech methods that are now accessible to everyone. As a result, finding security solutions continues to be a major challenge for researchers.

In recent years, researchers have worked to overcome intrusion detection difficulties in IoT networks. These studies employ a variety of methods, including machine learning (ML), semi-supervised learning, adaptive, heuristic, decision tree, and deep learning (DL). This section discusses relevant research on dealing with imbalanced data concerns in network-based intrusion detection, as well as other existing intrusion detection ideas for IoT networking settings [11].

Sharma et al. [12] discuss various intrusion detection system (IDS) approaches, including misuse-based, specification-based, and anomaly-based detection. The study evaluates the advantages and limitations of each method and provides recommendations for IDS deployment in sensor

Online: https://doi.org/10.64680/jisads.v3i2.59

networks. Additionally, the authors highlight several potential directions for selecting an appropriate IDS.

In [13], Wazid et al. present an IDS scheme for detecting routing attack that can be occurred in an IoT network-based environment called RAD-EI. The authors proposed two algorithms: an algorithm to detect the existence of routing attackers using the remaining energy of the node and its identity and algorithm to check and confirm these routing attackers. Then, the checked nodes are maintained in the blacklist and alarm messages are sent to normal nodes to notify them about these malicious nodes.

In [14], Moustafa et al. present an IDS based on a client-based system that exploits anomalies to identify an attacker known as E-Spion. Three-layered security was considered for improved security. However, the higher protection level resulted in greater overhead. In [15], Sun and Yu establish and suggest intrusion detection techniques for detecting the routing attack affecting the environment in Edge-based IoT (E-IoT).

In [16], Tkachenko et al. improve prediction accuracy for missing IoT data recovery by combining General Regression Neural Network (GRNN) with Successive Geometric Transformation Model (SGTM). However, experiments using publicly available datasets lacked performance comparison. Man et al. [17] propose the IDS model for the IoT environment, in which they employed the ULEACH clustering technique to increase node usage and performance. They also employed PSO to balance the efficiency of intrusion detection.

In [18], Venkatraman and Surendiran suggest hybrid IDS based on the studied multimedia files. To train the IDS model and identify intruders in IoT networks, they used internet resources as repositories. The acquired test findings were up to 99.06 percent accurate for recognizing various attacks in IoT settings such as denial-of-services (DoS), control hijacking.

In [19], Mohamed et al. describe the most recent IDSs for the IoT context, including their techniques, processes, and characteristics. Al-Hadhrami et al. in [20] explore several datasets for use in IoT ecosystems in order to evaluate various approaches for creating a smart environment. Following that, they demonstrated real-time data-gathering architecture for intrusion detection, allowing IDS for the IoT context to be reviewed and tested.

Yi et al. [21] developed an IDS for IoT environments that employs multi-CNN, a DL methodology, and a fusion method to detect intrusions. Multi-CNN displayed the intrusion detection problem and classified the intrusions using fusion. The authors achieved excellent accuracy for binary and multiclass classification on the NSL–KDD dataset. Mohamed et al. [22] present RDTIDS, a new IDS paradigm for IoT network settings. The model then utilized several classifiers based on DT and rules, such as JRip, Forest, and REP Tree. The authors tested their model on the CICIDS-2017 and BoT-IoT datasets. Jagadeesh and Raji [23] investigated and suggested a specification heuristic technique to determine the breadth of an intrusion in an IoT network. Keserwani et al. [24] presented a GWO–CSA–DSAE NIDS model for virtualized cloud networks (VCNs), in which GWO–CSA were utilized for feature selection while deep sparse auto-encoder (DSAE) was employed as a classifier.

Despite the fact that many current IDSs contributed significantly to attack detection, addressing data unbalancing concerns and selecting relevant characteristics to assist improved classification remains a difficulty.

3.2 The Energy Challenge

According to [25], electricity accounts for approximately 70% of total data center running expenses, thereby emphasizing the need to reduce energy use. Researchers have attempted to suggest answers in this field using a variety of techniques, including heuristics and algorithms.

The difficulty with these approaches is that they are not workload-specific, they are unable to handle workload fluctuations due to a lack of dynamicity, and they require previous knowledge of the workload in order to modify parameters. However, some researchers have attempted to tackle the problem using machine learning. Machine learning can manage fluctuating workload behavior since it is workload-specific, and it does not require workload specialization.

Green IoT aims to minimize the energy usage of IoT devices while also protecting the environment [26]. As a result, it is critical to use an efficient routing strategy for network routing. Multi-hop routing and clustering are two approaches that have been proposed to increase network performance and energy efficiency [27]. One or more intermediary nodes direct data to the base station in multi-hop routing. Direct data transfer would suddenly discharge the nodes if the source nodes are far from the base station; thus, multi-hop routing can minimize node energy usage [28].

Minimum transmission energy is a multi-hop routing approach in which each node directs data from other nodes in order to decrease the overall energy of the transfer [29].

Online: https://doi.org/10.64680/jisads.v3i2.59

The basic concept behind energy-aware IoT-based techniques is to save energy by utilizing load characteristics and VM consolidation technologies. We examine the energy-aware approaches employed by IoT applications to save energy within a Cloud Data Center (CDC).

Some researchers have employed the weighted sum [30], normalization [31], and normalizing [32] methods to arrive at a single best solution for tackling energy-aware IoT-based strategies in CDC. [33] introduce a new priority, power, and traffic-conscious VM placement algorithm that aims to reduce the energy use of incoming IoT traffic servers. Finally, the authors of [34] provide a safe technique for controlling the impact of IoT device requests on the network. Moreover, an energy-aware algorithm was intimately connected to the energy consumption model, and every energy-saving algorithm was reliant on a certain power model. The accuracy of the power model directly reflects the benefits and drawbacks of an energy-saving method. Specifically, in [35], the authors used deep learning technology to create a unique energy consumption model. Their approach considered 12 energy-related variables and used deep neural network architecture to create an energy consumption model. The authors of [36] present numerous power models, including mixed load and I/O-intensive demand. The authors of [37] introduce an energy-efficient work prioritization model in order to create a fairness job-scheduling algorithm in the CDC. Nonetheless, the suggested model in this study focused on the energy model, which was a threshold-based provisioning mechanism that was validated using a real-world workload.

4 Overview of the proposed intelligent hybrid IDS

Security characteristics and system performance are always tradeoffs in Cloud-IoT environments. Consequently, maximizing privacy protection while minimizing energy consumption is a major challenge that must be considered.

Therefore, the main objective of our work is to develop a system that can:

- Provide high security data by encrypting transferred information among IoT-cloud.
- Detect malicious actions and identify network attacks.
- Optimize energy consumption by integrating a sub-system that manages the utilization of IoT sensors and data centers.

The aforementioned objectives will ultimately lead to a global and comprehensive system for detecting and dealing with threats that impede the proper functioning of IoT-cloud networks. This work will then lead to best service management.

The first part of our work outlines a proper system model that includes various components which will be incorporated in an IoT-cloud system. This generalized model can be used in different scenarios depending upon the application requirements. The proposed architecture includes a model for ensuring data security and energy efficiency. In fact, for network detection attacks, we implement an IDS in the IoT-cloud environment. To reduce energy consumption, we deploy a dimension reduction strategy.

Figure 2 depicts the proposed architecture, illustrating an overview of our proposal.

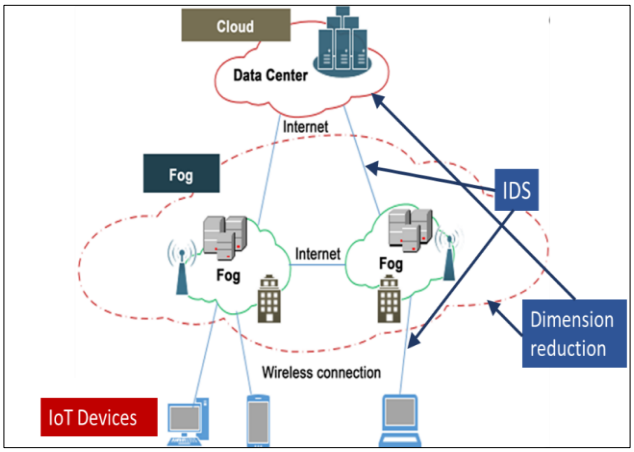

Figure 2. Overview of the proposed architecture

4.1 Intrusion Detection

An effective IDS must meet certain needs such as flexibility, real-time, and scalability. Many studies have been conducted to develop a next-generation IDS that meets all these needs. IDS is composed of five components, as illustrated in Figure 3 [2].

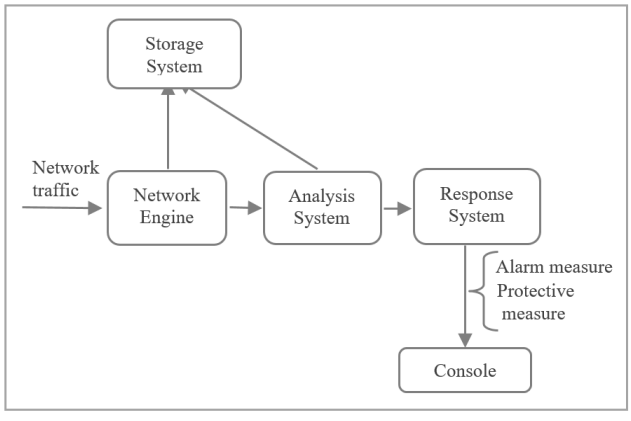

Figure 3. IDS's components

In this study, we focus on making the IDS more intelligent in detecting attacks by intervening inside its components. The primary role of Network Engine component is to analyze all network traffic (protocols, ports, subnets, etc.) and then, it interacts with the Analysis System, which is responsible for detecting intruders. The Analysis System consists of five modules: a pretreatment module, a rule knowledge base, a protocol analysis module, a data analysis module, and a secure communication module. Its function is to examine data from the Network Engine. When an intrusion is detected, the Response System must take the appropriate action of either an alarm or a defensive measure. These metrics are transmitted to the Console, so that they may be shown to users.

In this context, we propose the use of classification techniques to deal with the problem of intrusion detection. In fact, an intelligent analysis process improves the network's awareness of the environment's changes in order to enhance performance, mainly in relation to the latency and energy consumption metrics. Specifically, we applied a set of classification tools on a real IoT dataset to detect intrusion. Figure 4 shows the detailed steps of our proposal.

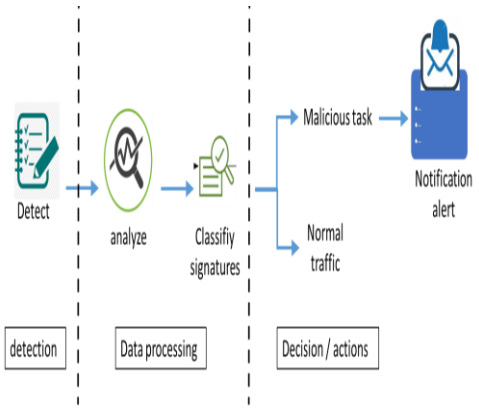

Figure 4. Overview of the proposed intelligent intrusion detection

We install an Intelligent Intrusion Detection System (IIDS) on the IoT-cloud network to monitor the whole network, oversee traffic, and analyze transferred data. After detecting a new attack, which is recognized by new signatures, the IIDS classifies the traffic as either normal or anomalous.

The proposed attack detection model may be used to compensate for the lack of a monitoring mechanism and to avoid potential threats. This activity is responsible for arranging the IIDS in order to acquire the basic information that will be utilized for the learning phase. The IIDS should be able to recognize network properties such as components, architecture, type, available and running services, operating systems, memory capacity, and even potential vulnerabilities.

After gathering the essential information, the IIDS analyzes and categorizes traffic flows as either normal or malicious. It can identify abnormal traffic by comparing it to regular traffic using a machine learning technique after analyzing a large amount of data. Then, it employs network audit technologies to produce alarms.

The proposed smart attack detection system consists of six steps handled by six layers.

5. System Design

5.1 Data Collection Layer

The data collection layer can be composed of cloud infrastructure monitors (e.g., AWS CloudWatch, Azure Monitor), network packet analyzers (e.g., Wireshark, Zeek), and Virtual Machine (VM) agents and hypervisor monitors. Its role is to capture all types of actions in the whole environment, such as network traffic, system logs, and user activities.

5.2 Data Preprocessing Layer

In this layer, we apply some algorithms in order to clean and transform raw data into meaningful formats. To improve machine learning performance and reduce dimension, common approaches are used which are feature extraction, data normalization and noise removal to filter redundant or irrelevant data. Feature extraction is used to apply some transformation to important characteristics in order to create more significant features. This technique is used also to reduce complexity and offer a straightforward representation of data, by treating each variable in feature space as a linear combination of important input variables. In the other hand, data normalization technique is used to scale data for consistent model input.

5.3 Detection Engine Layer

The role of detection engine layer is to analyze processed data to detect malicious activities, using artificial intelligence techniques, like Machine Learning (ML), Deep Learning (DL) or either Hybrid Models that combine ML and DL for optimized detection.

5.4 Decision-Making Layer

The purpose is to classify behaviors as normal or malicious and triggers responses. Furthermore, the goal of this step is to create classifiers that employ the characteristics

of the attack to distinguish attacks from regular data. Following this phase, the system will determine whether the input data constitutes an attack.

5.5 Response Layer

This agent will send alerts to notify system administrators. Besides, it can block malicious traffic. Moreover, it segregates affected VMs or services.

5.*6. Feedback and Learning Layer*

The main purpose of this layer is to continuously improve detection capabilities, by updating the detection model with new data. Furthermore, it adjusts detection strategies for evolving threats.

Figure 5 is a visual representation of the proposed IIDS model for IoT-cloud computing. It illustrates the six interconnected layers, highlighting data flow and security processes.

Our intelligent system has significant potential in attack detection, and it may be ideal for real-time applications in IoT-cloud computing.

5.7. Energy Efficiency

In the IoT-cloud environments, the phase of analyzing and treating heterogeneous data obtained from many sources is difficult and time-consuming. The management of these data is difficult without ignoring duplication, irrelevant, and unneeded data. Moreover, these environments consume a large amount of energy to provide efficient and reliable services to users. To tackle this issue, we aim to lower energy consumption in IoT–cloud systems by removing redundant data through artificial intelligence (AI) techniques.

AI methods help distinguish unnecessary information collected by IoT devices, preventing excessive demands on storage and computational resources. By filtering out duplicate or irrelevant data, the system avoids overloading devices, network connections, and servers, thereby supporting optimal IoT functionality. In this study, PCA (Principal Component Analysis) is applied as a feature extraction technique to examine its impact. To improve machine learning efficiency and reduce data dimensionality, two widely used strategies are employed: feature selection and feature extraction.

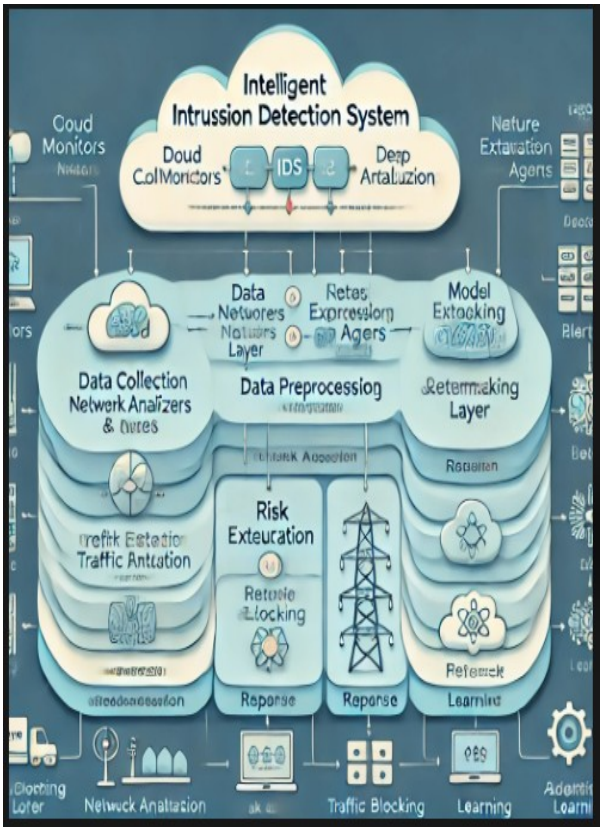

Figure 5. Visual Representation of the proposed IIDS model

Figure 5 illustrates techniques used in dimensionality reduction.

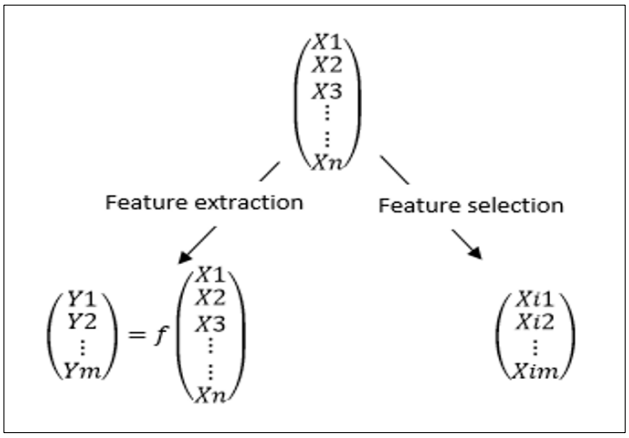

Figure 6. Feature selection vs. feature extraction

Feature selection is a dimension-reducing method used as an advanced step in machine learning. It is effective in removing unrelated data, lessening changes, increasing learning accuracy, enhancing predictor performance, improving understanding of results, providing faster and

lower-cost predictors, and emphasizing the most important process that generated the data [14, 15]. Feature selection is classified into three types: wrappers, filters, and embedding. Wrappers are superior to filters since they were optimized for the classifier employed. As a result, wrapper techniques have a high computational cost; thus, they are expensive and sluggish when employed for big features.

Feature extraction is the process of applying some transformation to important characteristics in order to create more significant features. To reduce complexity and offer a straightforward representation of data, we use feature extraction, which treats each variable in feature space as a linear combination of important input variables. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are the two primary feature extraction approaches that have been widely employed in various fields.

6. Experimental Setup

In this section, the evaluation of our contribution is outlined.

6.1 Simulation Parameters

As mentioned previously, we propose implementing an intelligent IDS within the IoT-cloud environment. Then, our goal is to evaluate our proposal by applying a set of classifiers on a real dataset [37][38] for intrusion detection.

To evaluate the performance of our proposed intelligent system, we have adopted a structured approach. Indeed, after collecting managed data, data preparation occurs. This step, which is sometimes referred to as "pre-processing," is the stage in which the data is cleaned and structured for the next stage of the information process. The goal of this preparation is to eliminate low-quality information that may be missing, redundant, or inaccurate, and to start creating information that will ensure the high quality of the selected intelligent model. In fact, the raw dataset is meticulously scrutinized for errors of any type. During this step, we convert the input data into a vectorizable matrix. After that, the classifier returns whether there is an attack in the network traffic. As a result, we apply the intelligent IDS by employing a number of classifiers in order to select the best one.

Numerous types of datasets have been used by researchers in the security domain to train and evaluate the performance of their systems. The most commonly used one is NSL-KDD dataset [39]. The dataset was extensively processed to eliminate duplication and/or missing records. The traffic distribution of the NSL-KDD dataset is shown in Table 1 [6]. The learned features were applied to the labeled test dataset to identify it as an attack or regular traffic.

Table 1. Traffic distribution of the NSL-KDD

Traffic	Training	Test
Normal	67343	9711
Dos	45927	7458
Probe	11656	2754
R2L	995	2421
U2R	52	200
Total Attack	58630	12833
Total Traffic	125973	22544

The used NSL-KDD dataset is pre-processed through the following steps:

- Step 1: Collecting and receiving data frames from the dataset in both train and test and having a column for labeling CSV files.
- Step 2: Merging all the collected data frames into one Data Frame (DF). Then, searching for missing, duplicated, and mistaken values in all columns, and dropping them from the DF.
- Step 3: Normalizing numeric data with a min-max scalar (between 0 and 1), adding a target column to the data frame, and then deleting constant features to select the best ones:
  - Initial number of features = 122,
  - Deleted features are Index (['num_outbound_cmds'], dtype='object'),
  - The number of features is now 121.

Afterwards, our goal is to reduce the number of the selected features while conserving the total variance at a great value as step 4:

- Step 4: Reducing the dimension of the selected features using Principal Component Analysis (PCA) to have the following final configuration:
  - Number of components = 10,
  - Total variance from PCA components = 0.8621156371639519.

To evaluate the proposed IIDS, we use several machine/deep learning algorithms in the context of intrusion detection, using the NSL-KDD dataset to choose the best one. Numerous metrics can be used in this test, such as accuracy (Ac), recall, detection rate (DR), positive predictive value (PP), and negative predictive value (NP). These factors are assessed by using true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

Online: https://doi.org/10.64680/jisads.v3i2.59

- A confusion matrix is a matrix that predicts the work of a tested classification method versus actual classification.

A confusion matrix compares predicted and actual class labels. Accuracy (Ac) represents the proportion of correct predictions and is defined mathematically in Eq. (1):

$$Ac = \frac{Tp+TN}{TP+TN+FP+FN} \quad (1)$$

- Recall (sensitivity) quantifies the proportion of actual attacks correctly detected and is defined as:

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

- Positive predictive (PP) value/Precision is the percentage of proximity between the predicted positive results compared to the value when the true condition is positive. Its mathematical formulation is as follows:

$$PP = \frac{TP}{TP+FP} \quad (3)$$

- F1 score represents the weighted harmonic mean of recall and precision. It is formulated as follows:

$$F1score = \frac{2*Recall*Precision}{Recall+Precision} \quad (4)$$

- Negative predictive (NP) value represents the ratio of the predicted negative values compared to the value when the true condition is negative. Its mathematical expression is as follows:

$$NP = \frac{TN}{FN+TN}$$

7. Discussion On The Results

7.1 Performance Evaluation for Dimension Reduction

In this subsection, we first discuss the used dataset. Then, we apply the PCA method as a feature extraction to this dataset to visualize and analyze the obtained results.

Among various data preprocessing techniques, principal component analysis (PCA) is employed in this study to reduce the dimensionality of a real-world dataset. PCA transforms the original features into a smaller set of principal components, minimizing data size while retaining critical information. As an unsupervised method, it performs this transformation without using class labels, making it suitable for datasets with limited or unlabeled data.

We conclude that reduction approaches improve the cloud of things' performance by removing superfluous data. Only the most critical and meaningful data are processed next.

Figure 7 shows the results of applying principal component analysis (PCA) for dimensionality reduction. The original dataset contained 449 features, which were reduced to 40 principal components after PCA, corresponding to an 89.08% decrease. This substantial reduction is expected to help lower resource usage and improve system efficiency in subsequent processing.

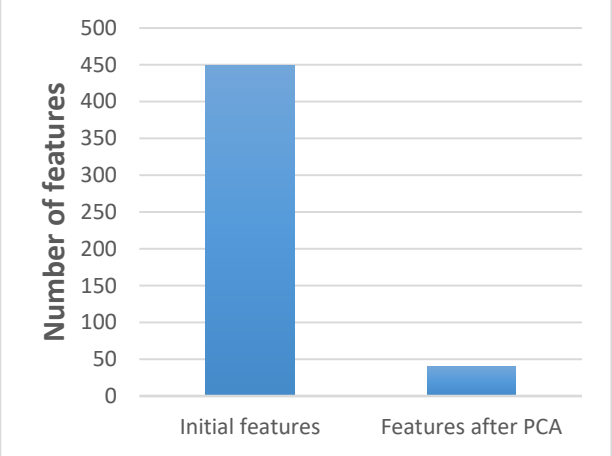

Figure 7. Number of features before and after applying PCA

The size of the dataset is shown in Figure 8. The initial zipped dataset is approximately 80MB, but after unzipping it, it becomes approximately 527MB. Without zipping, the new size of the dataset after PCA is approximately 28MB. As a result, we can deduce that using PCA has a significant influence on data size reduction. This fact of lowering dataset size verifies the prior conclusion of feature minimization.

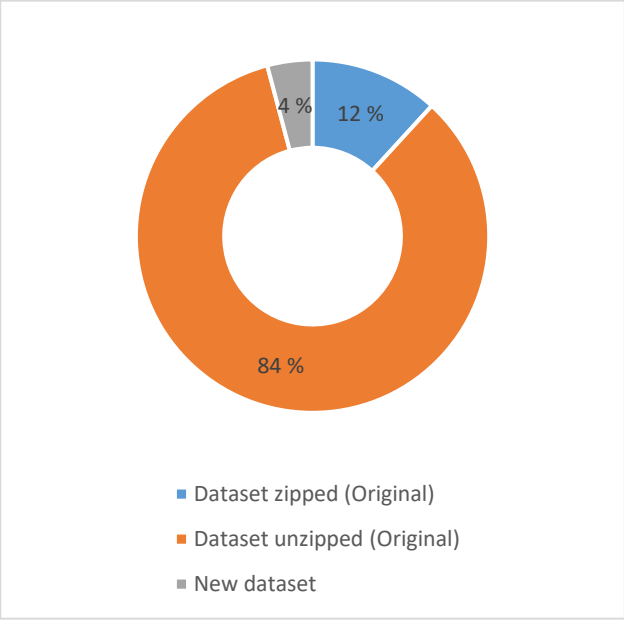

Figure 8. Dataset size

Online: https://doi.org/10.64680/jisads.v3i2.59

6.2 Performance Evaluation of the Intelligent IDS

The proposed intelligent intrusion detection system was evaluated against several machine learning and deep learning models using the NSL-KDD dataset. Table 2 presents the comparative results, measured in terms of accuracy, detection rate (DR), false positive rate (FPR), precision, recall, and F1-score. The evaluation highlights the relative strengths of each model in detecting network attacks.

Table 2. Performance Metrics

Model	Accuracy (%)	Detection Rate (%)	False Positive Rate (%)	Precision (%)	F1-Score (%)
Random Forest (RF)	96.5	95.8	2.3	96.0	95.9
SVM	92.4	90.7	4.1	91.5	91.1
Deep Neural Network	98.2	97.6	1.5	98.0	97.8
Hybrid IIDS (RF + DNN)	99.5	98.9	0.9	99.0	98.9

6.3 Simulation Results Analysis

We can deduce that the Hybrid IDS achieves the highest detection rate of 98.9%, meaning it effectively identifies most intrusion attempts. The Hybrid model also minimizes false alarms with a 0.9% FPR, critical for operational efficiency in cloud systems.

In fact, Deep Learning models, especially hybrid approaches, outperform traditional ML models, indicating better generalization.

6.4 Resource Utilization in Cloud

The simulation demonstrates that integrating intelligent models, particularly hybrid approaches, significantly enhances IDS performance in cloud environments by:

- Increasing detection accuracy
- Reducing false alarms
- Maintaining real-time detection speed

Table 3 shows the resource utilization in cloud.

Table 3. Resource utilization in Cloud

Model	CPU Usage (%)	Memory Usage (GB)	Detection Latency (ms)
Random Forest (RF)	45	2.3	150
SVM	40	1.8	170
Deep Neural Network	70	4.5	120
Hybrid IDS (RF + DNN)	75	5.0	**100**

6.5 Discussion of Simulation Results for Intelligent IDS in Cloud Systems

The simulation results for various Intelligent Intrusion Detection Systems in cloud environments highlight several important trends and insights:

a. Performance of Different Models

The simulation results show that DNN models are effective for learning complex attack patterns but may require more computational resources, potentially impacting cost and scalability in resource-constrained environments. Thus, RF is suitable for environments with limited computational resources. Furthermore, the suggested Hybrid IDS presents the highest accuracy (99.1%) and detection rate with 98.9%. These results suggest that combining Random Forest (RF) with Deep Neural Networks (DNN) leverages the strengths of both models.

Besides, the Low False Positive Rate (0.9%) ensures fewer false alarms, which is crucial in dynamic cloud systems where high false alarms can overwhelm security teams. In fact, our hybrid model balances detection accuracy and resource efficiency, making them ideal for cloud systems that demand real-time protection and scalability.

b. Trade-offs Between Detection Performance and Resource Consumption

Hybrid and DNN models outperform traditional ML models in detection performance but consume more CPU and memory. In other side, RF and SVM are lighter on resources but compromise on detection accuracy and false positive control.

Currently, cloud providers must balance detection efficiency with operational costs. High-performing models

like Hybrid IIDS are preferable for high-risk environments, while RF may suffice for lower-risk systems.

c. Detection Latency

Rapid response times are vital for mitigating fast-moving attacks like Distributed Denial-of-Service (DDoS) and zero-day exploits. Models with higher latency may not be suitable for time-sensitive detection tasks. That's why our Hybrid IIDS has the lowest detection latency (100 ms), enabling near real-time threat detection.

d. Scalability and Adaptability

Traditional models (RF, SVM) require regular retraining with new data, limiting their adaptability. However, deep learning models our Hybrid IIDS can adapt to evolving threats due to their ability to learn complex attack patterns.

In dynamic cloud environments, adaptability to new attack vectors is crucial. IIDS solutions must be scalable and capable of learning from new threats in real-time.

e. False Positive Rate (FPR)

High false positive rates in cloud systems can lead to alert fatigue and inefficient incident response. Models with low FPR are preferred for maintaining security team productivity. Hybrid IIDS and DNN significantly reduce false positives, minimizing unnecessary alerts.

Figure 6 shows that:

- Hybrid IIDS models offer the best balance of detection accuracy, low false positives, and response time, making them ideal for modern cloud systems.
- DNNs are also highly effective but may increase operational costs due to higher resource consumption.
- Random Forest models are suitable for cost-sensitive environments with moderate security needs.
- SVM is not recommended for large-scale cloud IDS due to its lower performance.

The evaluation results confirm the high effectiveness of the proposed intrusion detection approach in IoT–cloud environments. Across multiple classification metrics, the system demonstrates strong detection capability. When using the Random Forest (RF) classifier, the detection accuracy reaches 99%, while the false positive rate remains low at 0.00847, indicating reliable attack identification with minimal false alarms.

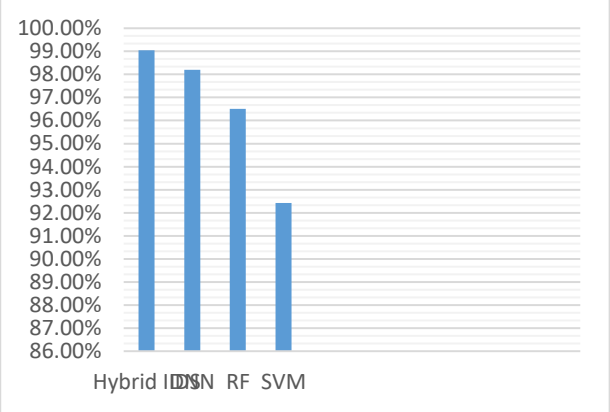

Figure 9. Accuracy comparison

Table 2 presents a comparative statistical analysis between RF and other classifiers, including Decision Tree, Gradient Boosting, Support Vector Machine, Artificial Neural Network, AdaBoost, Logistic Regression, Quadratic and Linear Discriminant Analysis, and Gaussian Naïve Bayes, as reported in [40]. The results show that RF consistently outperforms all competing methods in terms of precision, recall, and F1-score. Moreover, the favorable balance between true positives and false negatives highlights RF as the most effective classifier for attack detection in the proposed framework.

7. Conclusion

This work addresses critical security and performance issues in IoT–cloud systems, where the limited computational and storage capabilities of IoT devices restrict the deployment of complex security mechanisms. To overcome these constraints, lightweight and intelligent solutions are essential. Accordingly, an intelligent intrusion detection system (IIDS) is introduced, and multiple classification-based learning models are analyzed to determine their suitability for IoT environments. Furthermore, an encryption mechanism is integrated to provide essential security services, including confidentiality, integrity, and non-repudiation.

In terms of system efficiency, dimensionality reduction is employed as an effective strategy to lower energy consumption and enhance overall system performance by reducing data volume. This is achieved through artificial intelligence techniques such as feature selection and feature extraction. Experimental findings indicate that principal

Online: https://doi.org/10.64680/jisads.v3i2.59

component analysis (PCA) is particularly effective in minimizing data dimensionality without degrading detection accuracy. The proposed approach is experimentally validated, demonstrating its ability to improve both intrusion detection effectiveness and energy efficiency in IoT–cloud environments.

REFERENCES

[1] W. Ben Daoud, M. Rekik, A. Meddeb-Makhlouf, F. Zarai, and S. Mahfoudhi, "SACP: Secure access control protocol," in *Proc. Int. Wireless Communications and Mobile Computing Conf. (IWCMC)*, 2021, pp. 935–941.

[2] V. Kumar, A. Kumar, and D. Ditipriya, "UIDS: A unified intrusion detection system for IoT environment," *Evolutionary Intelligence*, vol. 14, no. 1, pp. 47–59, 2021.

[3] N. Hu, Z. Tian, H. Lu, X. Du, and M. Guizani, "A multiple kernel clustering-based intrusion detection scheme for 5G and IoT networks," *Int. J. Machine Learning and Cybernetics*, 2021.

[4] W. Ben Daoud, A. Meddeb-Makhlouf, and F. Zarai, "A model of role-risk based intrusion prevention for cloud environment," in *Proc. 14th IWCMC*, IEEE, 2018, pp. 530–535.

[5] W. Ben Daoud, M. S. Obaidat, A. M. Makhlouf, F. Zarai, and K. F. Hsiao, "TACRM: Trust access control and resource management mechanism in fog computing," *Human-centric Computing and Information Sciences*, vol. 9, no. 29, pp. 1–18, 2019.

[6] W. Ben Daoud and S. Mahfoudhi, "SIMAD: Secure intelligent method for IoT–fog environments attacks detection," *CMC–Computers, Materials & Continua*, vol. 70, no. 2, pp. 2727–2742, 2022.

[7] S. Mahfoudhi and M. Frehat, "Enhancing cloud of things performance by avoiding unnecessary data through artificial intelligence tools," in *Proc. 15th IWCMC*, 2019, pp. 1463–1467.

[8] S. Saraswat and H. P. Gupta, "Energy efficient data forwarding scheme in fog-based ubiquitous systems with deadline constraints," vol. 17, no. 1, pp. 213–226, 2020.

[9] Q. Duy, M. V. Ngo, T. Quang, T. Q. S. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," *Digital Communications and Networks*, vol. 5, no. 1, pp. 3–9, 2019.

[10] L. Goasduff, "Gartner says 5.8 billion enterprise and automotive IoT endpoints will be in use in 2020," 2019. [Online]. Available: https://gtnr.it/35hq94q (accessed Apr. 28, 2021).

[11] G. Kalnoor, "IoT-based smart environment using intelligent intrusion detection system," *Soft Computing*, vol. 25, no. 17, pp. 11573–11588, 2021.

[12] V. Sharma, I. You, and K. Yim, "BRIoT: Behavior rule specification-based misbehavior detection for IoT-embedded cyber-physical systems," *IEEE Access*, vol. 7, pp. 118556–118580, 2019.

[13] M. Wazid, P. Reshma, D. Ashok, K. Das, and J. J. P. C. Rodrigues, "RAD-EI: A routing attack detection scheme for edge-based Internet of Things environment," 2019.

[14] N. Moustafa, K. R. Choo, I. Radwan, and S. Camtepe, "Outlier Dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog," *IEEE Trans. Information Forensics and Security*, vol. 14, no. 8, pp. 1975–1987, 2019.

[15] T. Sun, "A formal verification framework for security issues of blockchain smart contracts," 2020.

[16] G. Ensemble, "An approach towards increasing prediction accuracy for the recovery of missing IoT data," 2020.

[17] M. Zhou, L. Han, H. Lu, and C. Fu, "Intrusion detection system for IoT heterogeneous perceptual network," 2020.

[18] S. Venkatraman and B. Surendiran, "Adaptive hybrid intrusion detection system for crowdsourced multimedia IoT systems," pp. 3993–4010, 2020.

[19] P. K. Kumar, K. Mahesh, M. C. Govil, E. S. Pilli, and P. Govil, "A smart anomaly-based intrusion detection system for IoT networks using GWO–PSO–RF model," *Journal of Reliable Intelligent Environments*, vol. 7, no. 1, pp. 3–21, 2021.

[20] Y. Al-Hadhrami and F. K. Hussain, "Real-time dataset generation framework for intrusion detection systems in IoT," *Future Generation Computer Systems*, 2020.

[21] Y. Li et al., "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion," *Measurement*, p. 107450, 2019.

[22] M. A. Ferrag, L. Maglaras, and A. Ahmim, "RDTIDS: Rules and decision tree-based intrusion detection system for IoT networks," pp. 1–14.

[23] M. J. Babu and A. R. Reddy, "SH-IDS: Specification heuristics-based intrusion detection system for IoT networks," *Wireless Personal Communications*, vol. 112, no. 3, pp. 2023–2045, 2020.

[24] P. K. Keserwani, M. C. Govil, and E. Shubhakar, "An optimal intrusion detection system using GWO-CSA-DSAE model," *Cyber-Physical Systems*, 2020.

[25] N. Hasan, R. N. Toma, A. Nahid, M. M. M. Islam, and J. Kim, "Electricity theft detection in smart grid systems: A CNN–LSTM based approach," 2019.

[26] Y. Chen, S. Hao, and H. Nazif, "A privacy-aware approach for managing the energy of cloud-based IoT resources using an improved optimization algorithm," 2021.

[27] M. Sadrishojaei, N. J. Navimipour, M. Reshadi, and M. Hosseinzadeh, "A new preventive routing method based on clustering and location prediction in mobile IoT,"

IEEE Internet of Things Journal, vol. 8, no. 13, pp. 10652–10664, 2021.
[28] J. Huang, Y. Hong, Z. Zhao, and Y. Yuan, "An energy-efficient multi-hop routing protocol based on grid clustering for wireless sensor networks," *Cluster Computing*, vol. 20, no. 4, pp. 3071–3083, 2017.
[29] M. Bagheri, V. Nurmanova, O. Abedinia, and M. S. Naderi, "Renewable energy sources and battery forecasting effects in smart power system performance," pp. 1–18.
[30] E. Parvizi and M. Hossein, "Utilization-aware energy-efficient virtual machine placement in cloud networks using NSGA-III," *Cluster Computing*, vol. 23, no. 4, pp. 2945–2967, 2020.
[31] X. Xu et al., "A computation offloading method over big data for IoT-enabled cloud–edge computing," *Future Generation Computer Systems*, 2019.
[32] F. Ruan, R. Gu, T. Huang, and S. Xue, "A big data placement method using NSGA-III in meteorological cloud platforms," 2019.
[33] S. Azizi, "Priority, power, and traffic-aware virtual machine placement of IoT applications in cloud data centers."
[34] M. Ghahramani et al., "RSS: An energy-efficient approach for securing IoT service protocols against DoS attacks," vol. 8, no. 5, pp. 3619–3635, 2021.
[35] Y. Liang, Z. Hu, and K. Li, "Power consumption model based on feature selection and deep learning in cloud computing scenarios," 2020.
[36] W. Lin, Y. Zhang, W. Wu, S. Fong, and L. He, "An adaptive workload-aware power consumption measuring method for servers in cloud data centers," *Computing*, 2020.
[37] M. Hussain, L. Wei, A. Lakhan, S. Wali, and S. Ali, "Energy- and performance-efficient task scheduling in heterogeneous virtualized cloud computing," *Sustainable Computing*, vol. 30, p. 100517, 2021.
[38] Canadian Institute for Cybersecurity, "NSL-KDD dataset," Univ. of New Brunswick, 2009. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html (accessed Jan. 10, 2026).
[39] S. Kavitha, N. U. Maheswari, and R. Venkatesh, "Network anomaly detection for NSL-KDD dataset using deep learning," *IT in Industry*, vol. 9, no. 2, pp. 821–827, 2021.
[40] R. Jablaoui, O. Cheikhrouhou, M. Hamdi, and N. Liouane, "Deep learning enabled intrusion detection system for IoT security," *EURASIP J. Wireless Communications and Networking*, vol. 2025, no. 1, pp. 1–18, 2025.
[41] W. H. Aljuaid and S. S. Alshamrani, "A deep learning approach for intrusion detection systems in cloud computing environments," *Applied Sciences*, vol. 14, no. 13, Art. no. 5381, 2024.
[42] S. Al-Otaibi et al., "AI-driven intrusion detection and lightweight authentication framework for secure medical sensor networks," *Scientific Reports*, vol. 15, Art. no. 31981, 2025.
[43] M. Alamri, N. Tariq, and M. Humayun, "Energy-efficient threat detection in IoT healthcare using AI and blockchain-enabled fog–cloud architecture," *Cluster Computing*, vol. 29, no. 2, pp. 1–20, 2026.