systems

MDPI

Article

Enhancing Cloud-Based Security: A Novel Approach for Efficient Cyber-Threat Detection Using GSCSO-IHNN Model

Divya Ramachandran [1], Mubarak Albathan [2], Ayyaz Hussain [3] and Qaisar Abbas [2,*]

[1] Department of Information Technology, PSNA College of Engineering and Technology, Dindigul 624622, Tamil Nadu, India; divya07i211@psnacet.edu.in
[2] College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia; mmalbathan@imamu.edu.sa
[3] Department of Computer Science, Quaid-i-Azam University, Islamabad 44000, Pakistan; ayyaz.hussain@qau.edu.pk
* Correspondence: qaabbas@imamu.edu.sa

Abstract: Developing a simple and efficient attack detection system for ensuring the security of cloud systems against cyberthreats is a crucial and demanding process in the present time. In traditional work, various machine-learning-based detection methodologies have been developed for securing the cloud network. However, those methodologies face the complications of overfitting, complex system design, difficulty understanding, and higher time consumption. Hence, the proposed work contributes to the design and development of an effective security model for detecting cyberthreats from cloud systems. The proposed framework encompasses the modules of preprocessing and normalization, feature extraction, optimization, and prediction. An improved principal component analysis (IPCA) model is used to extract the relevant features from the normalized dataset. Then, a hybrid grasshopper–crow search optimization (GSCSO) is employed to choose the relevant features for training and testing operations. Finally, an isolated heuristic neural network (IHNN) algorithm is used to predict whether the data flow is normal or intrusive. Popular and publicly available datasets such as NSL-KDD, BoT-IoT, KDD Cup'99, and CICIDS 2017 are used for implementing the detection system. For validation, the different performance indicators, such as detection accuracy (AC) and F1-score, are measured and compared with the proposed GSCSO-IHNN system. On average, the GSCO-IHNN system achieved 99.5% ACC and 0.999 F1 scores on these datasets. The results of the performance study show that the GSCSO-IHNN method outperforms the other security models. Ultimately, this research strives to contribute to the ongoing efforts to fortify the security of cloud systems, making them resilient against cyber threats more simply and efficiently.

Keywords: Internet of Things (IoT); smart city; intrusion detection system (IDS); cloud systems; security; data preprocessing and normalization; improved principal component analysis (IPCA); grasshopper–crow search optimization (GSCSO); isolated heuristic neural network (IHNN)

Citation: Ramachandran, D.; Albathan, M.; Hussain, A.; Abbas, Q. Enhancing Cloud-Based Security: A Novel Approach for Efficient Cyber-Threat Detection Using GSCSO-IHNN Model. *Systems* **2023**, *11*, 518. https://doi.org/10.3390/systems11100518

Academic Editors: Nuno Lopes and Joaquim Gonçalves

Received: 5 September 2023
Revised: 11 October 2023
Accepted: 14 October 2023
Published: 16 October 2023

1. Introduction

Cloud computing on the Internet of Things (IoT) emerged as a revolutionary paradigm, profoundly influencing a myriad of fields, including healthcare systems, military applications, education, and beyond [1,2]. Its allure originated from its inherent cost-efficiency and remarkable reliability, which allowed organizations to scale their operations with unprecedented flexibility. However, with the increasing reliance on cloud infrastructure, there emerged an ominous and ever-present threat of cyberattacks [3–5]. These nefarious assaults on digital infrastructure disrupt normal system operations, perpetrating malicious activities that compromise data integrity, confidentiality, availability, and privacy.

In response to this growing menace, the imperative to fortify the security of cloud networks has become paramount. Recognizing the urgency of safeguarding these systems against cyberattacks, the concept of the intrusion detection system (IDS) was conceived [6,7].

The role of an IDS is pivotal; it exists to identify and thwart network intrusions, serving as the vigilant guardian of cloud systems. An IDS shoulders the responsibility of not only repelling cyber threats but also upholding the integrity, confidentiality, availability, and privacy of cloud-based operations [8,9].

In the annals of cybersecurity, various soft-computing approaches have been devised as part of conventional efforts to establish effective IDS frameworks [10]. IDS systems typically fall into two main categories: those employing anomaly-based detection and those relying on signature-based detection methods to facilitate trust in communication within cloud networks [11]. Among the arsenal of security models, those rooted in artificial intelligence (AI) have gained prominence as they exhibit a propensity for delivering precise IDS capabilities. Recent research endeavors have underscored the preeminence of machine learning-based classification techniques for the prediction of network breaches. Integral to the development of IDS frameworks are optimization approaches [12–14], which serve as guiding lights in the selection of pertinent features for training and testing the classifier. For applications related to the prediction and detection of network intrusions, a spectrum of machine learning models exists, encompassing supervised, semi-supervised, and unsupervised techniques [15–17]. While these methods have exhibited efficacy, they are not without their shortcomings [18–20]. Common issues include the complexity of comprehension, limitations in handling massive datasets, protracted processing times, substantial storage requirements, and elevated error rates.

In light of these challenges, the focus of this research endeavor is to promote the development of a streamlined, user-friendly strategy to secure cloud systems from cyber threats. By harnessing the power of innovation, the aim is to bridge the gap between the burgeoning complexity of modern cyber threats and the need for efficient, comprehensible, and effective security solutions for cloud-based ecosystems. This paper introduces a novel approach, combining the strengths of improved principal component analysis (IPCA), grasshopper–crow search optimization (GSCSO), and isolated heuristic neural network (IHNN) to create an IDS model that not only simplifies the security landscape but also significantly enhances the resilience of cloud systems against cyber threats. The GSCSO-IHNN system can find valuable applications in both smart cities and cybersecurity to enhance security and threat detection. This research endeavors to align with the collective mission of fortifying cloud system security, rendering them impervious to cyber threats, while offering a streamlined and efficient approach to achieving this vital objective.

The motivation behind this research is to address the increasing sophistication of cyberattacks in the ever-evolving landscape of cloud computing. It aims not only to prevent attacks but also to fortify the security of stored and transmitted data in cloud systems. The proposed method is designed to efficiently identify intrusions in network datasets while minimizing computational complexity and maximizing operational effectiveness.

1.1. Major Contributions

The following are the main research goals of this work:

- To increase the quality of input network datasets, efficient data preprocessing and normalization operations to ensure they are performed to produce noise-free data for further processing.
- To obtain the relevant features used for predicting intrusions, an improved principal component analysis (IPCA) mechanism is used.
- To optimally select the features based on the best optimal solution, a hybrid grasshopper–crow search optimization (GSCSO) technique is employed.
- To identify whether the data flow is normal or intrusive, an isolated heuristic neural network (IHNN) machine learning classification model is implemented.
- To evaluate the performance and outcomes of the proposed GSCSO-IHNN security system, the most popular and publicly available benchmarking datasets are used.

This research proposes a framework for detecting intrusion in cloud computing environments. The chosen methods aim to address specific challenges in intrusion detection

within cloud computing environments. The adopted methods aim to improve data quality, reduce dimensionality, optimize feature selection, and enhance classification accuracy, ultimately strengthening the security infrastructure in cloud systems.

1.2. Paper Organization

The remainder of this paper is divided into the following sections: Section 2 presents the background on machine-learning algorithms used in the previous studies, and Section 3 is used to briefly describe the various optimization and classification techniques used to detect intrusions from cloud-based systems. It also investigates the advantages and disadvantages of existing techniques based on their characteristics and detection performance. Section 4 provides a comprehensive description of the proposed methodology, including its general workflow and algorithmic examples. Utilizing a variety of evaluation parameters, Section 5 verifies the performance and efficacy of the proposed detection system. Section 6 contains discussions and potential future directions are described, and finally, the overall paper is summarized in Section 7.

2. Background

Machine learning (ML) methods have significantly contributed to the enhancement of cloud security by enabling the detection and mitigation of various cyber threats and vulnerabilities. These ML techniques encompass a wide array of approaches, each tailored to address specific security challenges. However, it is important to note that while ML offers many advantages, it also comes with certain limitations and challenges that need to be considered in cloud security applications. Supervised learning methods, such as support vector machines (SVM) and random forest (RF), have been extensively used for intrusion detection and classification in cloud systems. SVM seeks to find the optimal hyperplane to separate normal from malicious activities, while RF leverages ensemble learning to improve detection accuracy. However, supervised methods heavily rely on labeled training data, which can be scarce and may not adequately represent the evolving nature of cyber threats in the cloud. Additionally, the accuracy of these models can be compromised when faced with adversarial attacks designed to deceive them. Unsupervised learning techniques, including K-Means clustering and DBSCAN, are valuable for identifying anomalies and patterns in cloud network traffic without the need for labeled data. They can uncover unusual behavior that may indicate security breaches. Nevertheless, these methods can produce false positives or miss subtle threats, and they often require careful tuning of hyperparameters to achieve optimal results. Scaling these techniques to handle large and complex cloud environments can also be computationally intensive.

Deep learning methods, such as convolutional neural networks (CNN), long short-term memory (LSTM) networks, and autoencoders, have shown promise in cloud security due to their ability to process sequential and high-dimensional data. CNNs are effective at analyzing network traffic patterns, while LSTMs excel in time-series data analysis. Autoencoders are used for anomaly detection by learning to reconstruct normal data patterns. However, deep learning models are data-hungry and require substantial computational resources for training, making them less suitable for organizations with limited data or computational capabilities. They also tend to be opaque, making it challenging to interpret their decision-making processes.

Ensemble methods, such as gradient boosting and stacking, improve detection accuracy by combining multiple machine learning models. While these approaches generally yield better results, they can be computationally expensive and may require extensive feature engineering to be effective. Feature selection and engineering techniques, like PCA and RFE, are employed to identify relevant features and reduce dimensionality in cloud security datasets. However, selecting the right features and transforming them appropriately can be a time-consuming and manual process.

Hybrid approaches, which combine ML methods with optimization algorithms, aim to improve both detection accuracy and efficiency. These approaches can be highly effective

but may require expertise in multiple domains and can be complex to implement and maintain. Reinforcement learning, although less common in cloud security, offers the potential for developing adaptive systems capable of making real-time decisions in response to evolving threats. However, it requires substantial training and may not be well-suited to all cloud security scenarios.

Bayesian methods, including Bayesian networks and classifiers, provide a probabilistic framework for modeling relationships in cloud security data. They aid in threat identification and risk assessment by considering uncertainty. Nonetheless, Bayesian models can become computationally expensive as the complexity of the network increases, and they may not always capture complex, nonlinear relationships effectively.

In brief, machine learning (ML) methods have revolutionized cloud security by enabling automated threat detection and mitigation. However, they are not without limitations, including the need for labeled data, potential susceptibility to adversarial attacks, computational demands, and challenges related to model interpretability. The choice of ML approach should be carefully considered based on the specific security task, available data, and computational resources, and often a combination of methods is required to achieve robust cloud security. As cloud security threats continue to evolve, ongoing research and innovation in ML techniques will be essential to stay ahead of cyber adversaries.

3. Literature Review

This section provides a literature review of current approaches to cloud intrusion detection and classification [21–23]. There is an examination of the pros and cons of each method according to their own operational characteristics, important features, and functional nature.

To ensure network security with a lower false alarm rate, Ravipatti et al. [24] created a novel intrusion detection algorithm. In this setup, machine learning categorization is used to keep an eye on potentially harmful network traffic. Here, the KDD Cup'99 dataset is used to verify the quality of our work in terms of precision, recall, and recall error rate. However, the system model may be incomprehensible since it does not use specialized optimization and classification techniques for spotting anomalies. An exhaustive study of intrusion detection and prevention methods was published by Khraisat et al. [25]. It includes the methodologies of statistical techniques, knowledge-based techniques, and machine learning techniques. Moreover, it suggested some of the recent and popular intrusion datasets for validating the performance of IDS [26], which include the following:

- DARPA/KDD Cup'99.
- CAIDA.
- NSL-KDD.
- ADFA-LD/WD.
- CICIDS 2017.

According to the findings of this research, unsupervised machine learning approaches outperform their supervised counterparts in terms of detection accuracy. By combining the features of cuckoo search (CS) and particle swarm optimization (PSO), Ghosh et al. [27] created a hybrid optimization approach for identifying intrusions in network datasets. This work aims to improve the effectiveness of attack detection while reducing the complexity of the classifier through feature optimization prior to detection. In addition, the optimized feature set was used in conjunction with a number of machine learning-based classification approaches, including linear regression (LR), adaBoost (AB), and random forest (RF), to anticipate potential networking threats. Fewer resources were used, less time was invested, and detection proficiency was high as a result of this effort. Using additional high-dimensional datasets, however, it was unable to demonstrate the effectiveness of this detection approach.

A trust-based intrusion detection and classification (TIDAC) system was developed by Chkirbene et al. [28] to better identify abnormalities. Here, the filter and wrapper methods were used to choose the characteristics for enhancing the classification procedure. After

that, a soft combination mechanism was implemented to evaluate the credibility of nodes' actions. Features are chosen and trained, initial decisions are made, and then all of these decisions are merged into this framework. The incursions were also detected with the use of two supervised learning mechanisms: naive Bayes (NB) and online average one dependent estimator (AODE). However, this approach was limited by significant time expenditure, difficulty in understanding, and increasing complexity in calculations. A univariate ensemble-based classification model for network intrusion detection in cloud systems was developed by Krishnaveni et al. [28]. The goal of this study was to determine the optimal characteristics for enhancing classifier detection efficiency. In addition, it made good use of a voting system to differentiate between benign and malicious forms of communication. In addition, a paired *t*-test was run to verify the reliability of the results obtained. The main advantages of this study were its high detection accuracy, high performance, low error rate, and low dimensionality of features. The main drawback of this work was the longer training period needed for the features.

To determine the best method for detecting intrusions, Kanimozhi and Prem Jacob [29] examined the effectiveness and performance of several machine learning algorithms. Precision, accuracy, recall, f-measure, and error rate are used to verify the model's performance; ANN, RF, KNN, SVM, AB, and NB are all included. Using a model of hybrid semantic deep learning architecture, Prabhakaran et al. [11] improved cloud system security against malicious network attacks. This study combines LSTM, CNN, and SVM models into a single framework called hybrid semantic deep learning (HSDL). In addition, the recommended detection system's resilience and accuracy were improved with the use of the crossover mine blast optimization approach. However, this approach has significant limitations, including higher time requirements, processing overhead, and diminished effectiveness. An effective machine learning-based intrusion detection system (IDS) architecture was presented by Aldallal et al. [30] to bolster cloud data security. In this case, the GA-integrated SVM method was used to boost the system's overall detection and safety capabilities. Both high detection accuracy and low FPR are primary advantages of this method. Using a distributed, multi-agent-based IDPS, Javadpour et al. [8] analyzed both typical and unusual patterns in network traffic. The goal of this effort was to make cloud-based IoT systems more resistant to cyberattacks.

To find cyberattacks in the cloud, Geetha et al. [26] used the Fisher kernel-based principal component analysis (FKPCA). The primary goal of this effort was to improve convergence times, precision, overfitting, and performance. In this case, the low-overhead deep learning BiLSTM classification method is employed to detect intrusions. In order to identify intruders in fog systems, Kumar et al. [31] devised an ensemble learning approach. Building an IoT network that is both scalable and secure against contemporary assaults is the primary focus of our study. In this case, the random forest (RF) method is employed to provide the Internet of Things with a more robust layer of security. Preprocessing, feature mapping, data imputation, optimization, and classification are the five phases that make up this system.

Kilincer et al. [32] suggested a min–max normalization model for preprocessing the given datasets to improve the classification accuracy. After that, machine learning-based models were employed to predict the normal and abnormal attacking flows with minimal mis-prediction outcomes. Asif et al. [33] deployed a map-reduce model for developing an effective IDS for increasing the security of IoT networks. Shaji et al. [34] discussed overall monitoring and data gathering in cyberphysical systems that are performed by supervisory control and data gathering systems, which are the primary targets of attackers in order to make cyberspace applications unworkable. The goal of this study was to strengthen the network's defenses against hackers. The effectiveness of IDS detection has also been verified using various data categorization algorithms. This study comprised a literature review and concluded that the existing research focuses mostly on developing effective IDS models to protect computer networks against exploits and other forms of cybercrime. A high false alarm rate, a lengthy training period, complicated computations, and substantial processing

overhead were all challenges. Because of this, the study encourages the creation of a hybrid optimization-integrated machine learning classifier for the purpose of identifying attacks on cloud-based infrastructure.

As highlighted in Table 1 of the literature review, the existing techniques exhibit significant shortcomings, such as heightened latency, reduced detection efficacy, elevated energy consumption, and prolonged processing durations. To address these challenges, this research endeavors to formulate an effective intrusion detection method aimed at enhancing the security of IoT networks.

Table 1. A literature review of several research works related to intrusion detection and attack classification in IoT (Internet of Things) networks.

Cited	Purpose	Methodology	Results	Limitations
[24]—Ravipatti et al.	Develop intrusion detection algorithm with lower false alarms	Machine learning using KDD Cup'99 dataset	Achieved precision, recall, recall error rate	Lack of specialized optimization, model complexity
[25]—Khraisat et al.	Conduct exhaustive study of intrusion detection methods	Overview of statistical, knowledge-based, machine learning using several intrusion datasets, including DARPA/KDD Cup'99, CAIDA, NSL-KDD, ADFA-LD/WD, and CICIDS 2017.	Unsupervised ML outperforms supervised	Not mentioned
[27]—Ghosh et al.	Develop hybrid optimization approach for intrusion identification	Combine cuckoo search and particle swarm optimization with ML	Reduced resource usage, high detection proficiency	Effectiveness not demonstrated with high-dimensional data
[28]—Chkirbene et al.	Develop trust-based intrusion detection and classification system	Feature selection, supervised learning (NB, AODE)	Significant time and complexity	Prolonged processing durations
[28]—Krishnaveni et al.	Create univariate ensemble-based classification model	Optimal feature selection, voting system	High detection accuracy, low error rate	Longer training needed
[29]—Kanimozhi and Prem Jacob	Examine machine learning algorithms for intrusion detection	Evaluate model performance using various metrics	Not mentioned	Reduced detection efficacy
[11]—Prabhakaran et al.	Improve cloud system security with hybrid semantic deep learning	Combined LSTM, CNN, SVM	Enhanced detection, higher time requirements	Increased processing overhead
[30]—Aldallal et al.	Present effective ML-based IDS for cloud data security	Integrates GA and SVM	High detection accuracy	Reduced detection efficacy
[8]—Javadpour et al.	Analyze network traffic patterns using distributed multi-agent IDPS	Focus on IoT systems	Not mentioned	Heightened latency
[26]—Geetha et al.	Use Fisher kernel-based PCA to detect cyberattacks in the cloud	Improve convergence times, precision	Not mentioned	Prolonged processing durations
[31]—Kumar et al.	Devise ensemble learning approach for fog systems and IoT networks	Employ random forest for robust IoT security	Not mentioned	Reduced detection efficacy
[32]—Kilincer et al.	Suggest min–max normalization model for dataset preprocessing	Machine learning for normal and abnormal flow prediction	Minimal mis-prediction outcomes	Reduced detection efficacy
[33]—Asif et al.	Deploy map to reduce model for effective IoT network security	Not mentioned	Not mentioned	Prolonged processing durations
[34]—Shaji et al.	Strengthen network defenses in cyber-physical systems	Verify IDS detection with data categorization	Challenges of high false alarms, lengthy training, computational complexity	Reduced detection efficacy
[35]—Dua et al.	Mitigate network security vulnerabilities due to network expansion	Attribute selection and ensemble classifier	High accuracy	Dataset dependency, generalization and scalability
[36]—Mann et al.	Hybrid clustering algorithm for grouping GPS coordinates	K-means clustering and BIRCH (balanced iterative reducing and clustering using hierarchies)	Effectiveness	Clustering large unsupervised cab datasets

4. Materials and Methods

4.1. Proposed Framework

Cloud computing is a modern and more convenient way to store, access, and manage data. It is built on the idea of efficiently allocating resources, which ensures efficient computing, cost-effectiveness, scalability, and a high-quality service. However, with the increasing cyberattacks, it is not enough to prevent them. It is equally important to strengthen the security infrastructure to protect both stored and transmitted data. Vulnerabilities in cloud management can lead to disruptive attacks and data loss. This section proposes an intrusion detection mechanism that can safeguard cloud systems. The research uses innovative methodologies tailored for the efficient identification of network intrusions within datasets. It achieves this objective by mitigating computational complexity and optimizing operational efficiency. The research aims to secure cloud-based infrastructure and mitigate the challenges posed by the dynamic realm of cloud computing and the escalating sophistication of cyberattacks. This requires a multifaceted approach that synthesizes resource efficiency, state-of-the-art intrusion detection techniques, and an unwavering commitment to data security. The steps in the proposed system's workflow are shown in Figure 1.

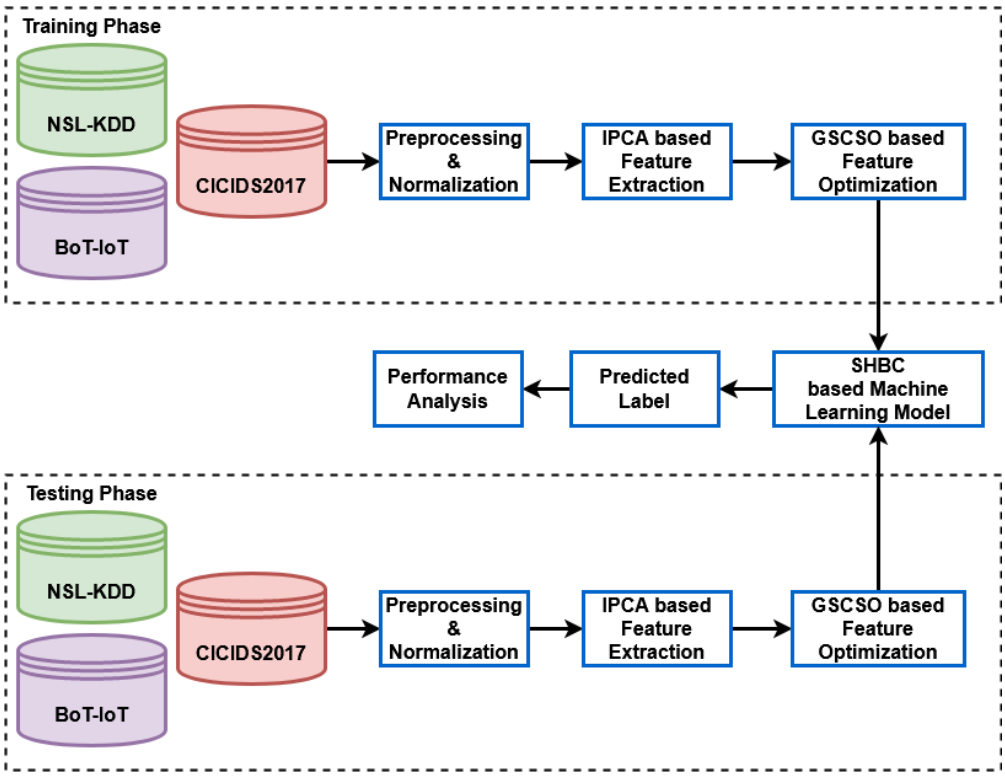

Figure 1. A systematic flow diagram of the proposed system.

(1) Dataset collection;
(2) Preprocessing and normalization;
(3) Feature extraction;
(4) Hybrid feature optimization;
(5) Machine learning-based classification;
(6) Performance evaluation.

The system commences with the acquisition of well-established and emerging network intrusion datasets, including NSL-KDD, BoT-IoT, KDD Cup'99, and CICIDS 2017. These datasets are foundational for the development and operationalization of the system. However, it is noteworthy that these benchmark datasets often present certain challenges, notably unbalanced attribute distributions, which can potentially compromise the accuracy

of classification outcomes. Therefore, to ensure the reliability of subsequent analyses, it is imperative that a series of preprocessing and normalization procedures be diligently applied. Preprocessing encompasses a multifaceted approach, including noise reduction, data rebalancing, identification of missing data fields, and standardization of attribute values. These preprocessing steps collectively serve the vital function of enhancing the overall quality and coherence of the datasets. By reducing noise and addressing data imbalances, the integrity of the data is safeguarded, laying a robust foundation for further analysis.

Subsequently, a feature extraction model is employed, primarily relying on the improved principal component analysis (IPCA). Feature extraction is a pivotal stage in the data processing pipeline, aimed at isolating pertinent information and reducing dimensionality. IPCA, a well-established technique, is leveraged for its ability to mitigate overfitting issues and also the impact of correlated features [37]. Through IPCA, the most salient attributes are distilled from the data, ensuring that only the most informative elements are retained for subsequent phases. IPCA is designed to reduce the dimensionality of data while retaining relevant information. This leads to more accurate intrusion detection by focusing on the most informative features. This addresses challenges faced by prior research by utilizing more effective feature selection and extraction methods leading to higher accuracy rates [38].

Furthermore, the process incorporates a unique dimension of optimization in the form of a hybrid approach, aptly named grasshopper and crow search optimization (GSCSO). This hybrid method plays a pivotal role in feature selection, a task of paramount importance in intrusion detection [39]. Notably, the selection of optimal features is instrumental in balancing the computational complexity of the system while ensuring the efficacy of the detection mechanism. Given that training and testing classifiers on extensive datasets can be time-consuming, feature optimization becomes a critical aspect of enhancing the system's efficiency. This addresses issues faced by traditional feature selection methods, which struggle with large datasets or fail to balance accuracy and computational complexity. It also addresses the resource constraints in cloud environments where traditional intrusion detection systems struggle to efficiently utilize available resources [40].

To complement these processes, the framework integrates a state-of-the-art machine learning model known as the isolated heuristic neural network (IHNN). IHNN serves as the cornerstone for determining the nature of data flows, categorizing them as benign or potentially malicious. This classification capability is of paramount importance in intrusion detection systems, as it empowers the system to make informed decisions about network activity [41]. IHNN's distinctive advantage lies in its expeditious and efficient operation, characterized by high convergence rates and detection accuracy. Conventional rule-based or signature-based methods are not as effective in identifying novel and sophisticated attacks. They also suffer from delays in processing and decision-making, which can impact their effectiveness in real-time scenarios [42].

This comprehensive approach holds significant promise in the realm of cloud intrusion detection. By addressing the challenges posed by unbalanced datasets and optimizing feature selection, it endeavors to enhance the system's ability to accurately and swiftly identify network intrusions within cloud-based infrastructure. The amalgamation of data preprocessing, feature extraction, hybrid optimization, and machine learning classification underscores the potential to bolster cybersecurity in contemporary cloud computing environments.

4.1.1. Preprocessing

In this work, various intrusion benchmark datasets like CICIDS 2017, BoT-IoT, and NSL-KDD are collected as inputs for system implementation. Normally, the original datasets comprise incomplete and missing fields of information, which affects the performance of classification and detection efficiency. Therefore, preprocessing and normalization

operations are essential before classification. The overall process of preprocessing step is presented in Algorithm 1. In this model, the distance is initially estimated as shown below:

$$IData_i^d[x,n] = \sqrt{(IData_i[k] - IData_i[x+n])^2} \quad (1)$$

where $IData_i$ is the incomplete data, $n = 1, 2, 3, \ldots, a-1$; $IData_i^d$ denotes the distance. Based on this, the minimum distance is estimated for all attributes exist in the dataset by using the following model:

$$IData_i{}^m[x] = min(IData_i^d[x,:]) \quad (2)$$

Based on this model, the minimum distance value is estimated and equated with the onset value for the particular attribute. Consequently, the data are normalized, and missing data are assigned for generating the filtered dataset.

Algorithm 1 Preprocessing and Normalization

Input: Incomplete data $IData_i$ with Feature dimension X;
Output: Normalized data FD_i;
Step 1: Initially, the distance is estimated according to the dimensionality of the given data.
for n = 1 *to* X
Estimate the distance $IData_i^d[x,n]$ as shown in Equation (1).
end for;
Step 2: Then, the minimum distance is estimated for all attributes in the dataset.
for n = 1 *to* X
Estimate the distance $IData_i{}^m[x]$ as shown in Equation (2).
end for;
Step 3: After that, the minimum value is estimated and compared.
for n = 1 *to* X
if $IData_i{}^m[x] < G_n$
if $IData_i[n,x] == null$
$IData_i[n,x] = IData_i{}^m[x]$
end if
else
$IData_i[n,x] = mean(ID_{atai}[n,:])$
end if
end for;
Step 4: Moreover, the redundant attribute fields are eliminated.
for x = 1 *to* X
//The repeated elements *repeatedElements*(.) in the particular attribute are computed.
if $rE(IData_i[x,:]) > 2$ and $rE(IData_i[x,:]) \leq 4$
$index_r = find(rE(IData_i[n,:]))$
$I[n, index_r] = Standard\ Deviation(IData_i[x,:])$
else if $rE(IData_i[x,:]) > 4$
$IData_i[x,n] = skewness(IData_i[x,:])$
end if
end for
Step 5: Based on the above calculations, the final normalized dataset is produced as the output.
for $n = 1$ *to* X
$IData_{in}[x,:] = IData_i[n,:]$
end for

4.1.2. Improved Principal Component Analysis (IPCA)

After obtaining the filtered dataset, an IPCA technique is deployed to extract the most appropriate features for increasing the presentation of the classifier. It is one of the most popular and efficient techniques widely used in many application systems due to the benefits of reduced overfitting, high performance, and elimination of correlated features. During this operation, the filtered data FD_i are considered as the input for processing, and the extracted features $ExaF_{Data}$ are produced as the output. Here, the center of data matrix Cen_i is calculated for each characteristic of the dataset as a function of the overall size of the sample.

$$Cen_i = \frac{1}{S}(FD_i^j \left(FD_i^j\right)^G) \quad (3)$$

where S is the number of samples; j indicates the iterations; Cen_i is the centered data matrix. According to the data dimensionality D_s, the eigen vector and eigen values are estimated by using the following models:

$$[{V_{Ei}}^i, E^i] = eigen(Cen_i, D_s) \tag{4}$$

where ${V_{Ei}}^i$ indicates the eigen vector, and E^i is the eigen values. Finally, the extracted set of features are in the following form:

$$ExaF_{Data} = ({V_{Ei}}^i)^T Im_D^i \tag{5}$$

The stages of the algorithm used in the PCA-based feature extraction approach are presented in Algorithm 2 as:

Algorithm 2 Principal Component Analysis

Input: Imputed data FD_i;
Output: Extracted features $ExaF_{Data}$;
Step 1: Initially, the center of data matrix is estimated for all the attributes in the filtered data by using Equation (3).
Step 2: According to the dimensionality of data, both the eigen vectors and values are computed for the center of matrix by using Equation (4).
Step 3: Based on the estimated values, the final complete set of results consists of feature extractions from the dataset of $ExaF_{Data}$;

4.1.3. Grasshopper—Crow Search Optimization (GSCSO)

The proposed research introduces an innovative hybrid optimization technique, called grasshopper–crow search optimization (GSCSO), to discern the most pertinent features for classifier training and testing after feature extraction. In the realm of intrusion detection systems (IDS), various bio-inspired and nature-inspired optimization methodologies have been utilized to solve intricate problems and obtain optimal solutions. However, these methods often face challenges such as sluggish convergence rates, constrained diversity in solutions, meager exploitation capabilities, and an unwieldy number of iterative processes. To overcome these issues, the GSCSO approach integrates highly efficient and widely recognized optimization techniques to facilitate the selection of optimal features from the pool of extracted feature data. The GSCSO approach offers several benefits, including an optimized subset of features, high proficiency and accuracy, efficient avoidance of local optima, and minimal iterative steps to reach optimal solutions. The merits of the GSCSO approach are manifold:

a. Optimized subset of features: GSCSO meticulously identifies a subset of features that contribute most effectively to the classification task, enhancing the efficiency of intrusion detection.
b. High proficiency and accuracy: By strategically selecting features, GSCSO significantly bolsters the accuracy and proficiency of intrusion detection systems, reducing the likelihood of false positives and negatives.
c. Efficient avoidance of local optima: The GSCSO method adeptly navigates solution spaces, evading the trap of local optima, and is particularly adept at exploring diverse solution landscapes.
d. Minimal iterative steps to reach optimal solutions: GSCSO streamlines the process of identifying the most suitable feature set, achieving optimal results with fewer iterative steps compared to conventional optimization techniques.

To explain the workings of this method, an initial population is generated after establishing the search space and relevant parameters. Then, fitness functions for the searching agents are computed, and the average fitness value is determined to identify the most favorable solution. Optimal threshold values are derived for each component based on the computed value. These threshold values are instrumental in selecting features for both the training and testing phases of the classifier, thereby improving the overall performance and

efficiency of the intrusion detection system. The working model of the proposed GSCSO technique is shown in Figure 2 and explained in Algorithm 3.

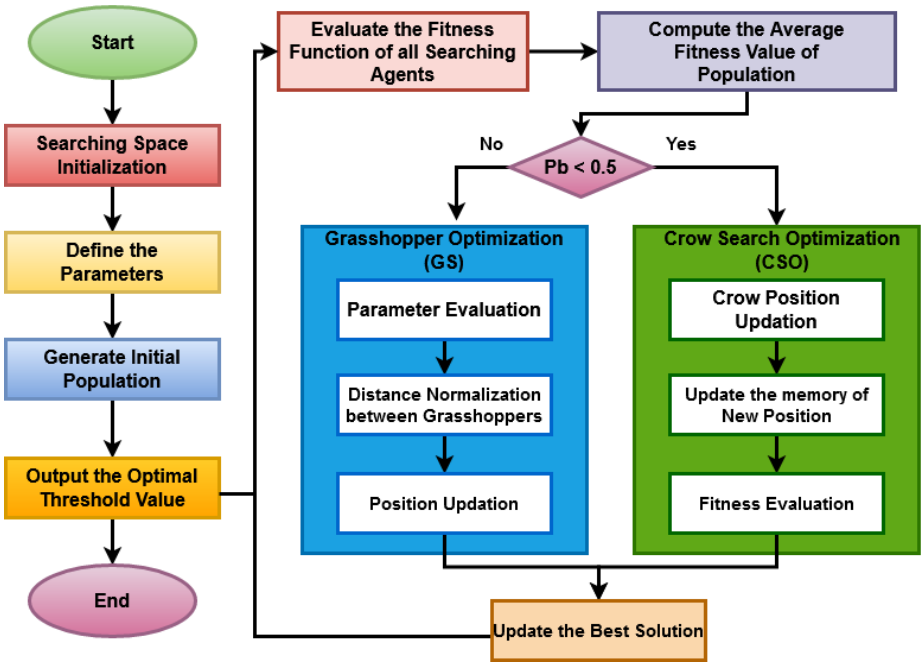

Figure 2. Working flow of GSCSO.

In this technique, the input parameters such as dimension d, random population P, and fitness function for each solution $p_i = 1, 2, \ldots N$ are initialized at first. After that, each solution is transformed into the binary value according to the thresholds $\rho \in [0, 1]$ as shown below:

$$p_i^{k+1} = \begin{cases} 1 \; if \frac{1}{1+e^{-p_i^k}} > \rho \\ 0 \; Otherwise \end{cases} \tag{6}$$

Based on this function, the particular elements having the binary value of 1s are chosen for representing the selected features. Then, the minimal objective function is estimated as shown below:

$$f(p_i^k) = \varepsilon Err_{p_i^k} + (1 - \varepsilon) \frac{|p_i^k|}{|NF|} \tag{7}$$

where $Err_{p_i^k}$ indicates the error rate; p_i^k denotes the length of subset; NF is the total number of input features. Consequently, the constant parameters $\varepsilon \in [0, 1]$ and $(1 - \varepsilon)$ are initialized for balancing the number of features according to the error rate of classifier. Moreover, the probability is estimated for each function as represented below:

$$Pb_i = \frac{f_i}{\sum_{i=1}^{PS} f_i} \tag{8}$$

where Pb_i is the probability value; PS denotes the size of population. If the value of probability Pb_i is greater than 0.5, the GS optimization is performed; otherwise, the CSO is performed. Moreover, the fitness value is computed for the updated solution, and based on this, the best optimal solution is identified. Then, this process is repeated until reaching the stopping criterion k_{max}.

Algorithm 3 GSCSO-based Feature Selection

Begin
Initialize the size of population PS, and maximum number of iterations k_{max};
Initialize the input parameters of grasshoppers as,
$GS_i = 1, 2, \ldots PS$;
Then, the fitness value is determined for the initial grasshoppers.
Initialize the mth magnitude of GS objective function $\hat{G_m}$ and the best fitness value B_{fit}
While $(k < k_{max})$
Estimate the fitness value according to the boundary value.
The position and best fitness functions are updated.
Then, the center controlling Cen_{con} parameter is computed as follows:
$Cen_{con} = Cen_{max} - h\frac{Cen_{max} - Cen_{min}}{H}$
//Where Cen_{max} and Cen_{min} are the maximum and minimum values correspondingly, h is the current iteration, and H denotes the cumulative number of iterations.
For i = 1 to n
The probability value is estimated by using Equation (8).
If $(pb > 0.5)$
The GS optimization is performed based on updating of the position.
Else
The CSO is performed based on updating of the position.
End if;
End for;
The probability value is restored.
The fitness function is computed for identifying the best optimal solution.
$k \leftarrow k + 1$;
End while
End;

4.1.4. Isolated Heuristic Neural Network (IHNN)

The features are then provided to the classifier for training and testing activities, where they contribute to an accurate label prediction. As the step in IDS responsible for determining whether a given flow is benign or malicious, categorization is crucial. In the previous studies, network intrusions or abnormalities were detected using a variety of machine learning-based categorization techniques. Neural networks (NN), support vector machines (SVM), naive Bayes (NB), multilayer perceptrons (MLP), LR, RF, and DT are some of the most commonly used methods for predicting attacks on cloud network security. However, it has issues with overlapping, being computationally costly and inefficient, and being unable to handle huge dimensional datasets. As a result, the suggested work aids in the creation of a new IHNN model for classifying the flow of data as either normal or an attack, based on the best attributes. The IHNN is a machine learning classification approach widely used to prevent attacks on cloud infrastructure. Training and testing procedures take advantage of the optimized set of features.

In this technique, the feature vectors and its weight values are initialized at first based on the following models:

$$FV = (fv_1, fv_2, \ldots, fv_n) \tag{9}$$

$$\omega = (\omega_1, \omega_2, \ldots, \omega_n) \tag{10}$$

where $fv_1, fv_2 \ldots fv_n$ are the feature vectors obtained from the optimized feature set; n is the total number of features; $\omega_1, \omega_2, \ldots, \omega_n$ are the weight values. After that, the smoothing parameter is computed to trigger the activation function as represented below:

$$f(FV) = exp\left(\frac{(\omega_i - fv)^k - (\omega_i - fv)}{2 \times \partial^2}\right) \tag{11}$$

where k indicates the current iteration; ∂ denotes the smoothing parameter. Moreover, the multiclass prediction is enabled by using the following equation:

$$(y_i x class_i x DF_i) > \left(y_j x class_j x DF_j\right) \qquad \forall i \neq j \tag{12}$$

where y_i and y_j are the preceding prospect; $class_i$ and $class_j$ are the misclassifying object that belongs to classes i and j; DF_i and DF_j are the probability density function of classes i and j.

Moreover, the Bayes rule is constructed for each class i and j by using the following model:

$$B_i(FV) > B_j(FV) \qquad \forall i \neq j \tag{13}$$

By using this rule, the best classification decision is made according to the probability density function as represented below:

$$B_i B_i(FV) = \frac{1}{s_i \times \partial} \sum_{l=1}^{s_i} \frac{FV - fv_{ir}}{\partial} \tag{14}$$

where fv_{ir} is the input features of lth training input. Moreover, the Euclidean distance is computed to train the samples, and the Gaussian function is utilized as the weight function, which is estimated by using the following model:

$$B_i(FV) = \frac{1}{s_i \times \partial} \sum_{l=1}^{s_i} exp^{\frac{(FV - fv_{il})^2}{\partial^2}} \tag{15}$$

According to this model, the classifier predicts the output data flow as normal or intrusion with minimal training and testing time.

$$\hat{\mathrm{W}}_{\mathrm{j}} = \frac{\hat{\mathrm{W}}_{\mathrm{j}}}{\sum_{\mathrm{r}=1}^{\hat{\mathrm{N}}} \hat{\mathrm{W}}_{\mathrm{r}}} \; \forall \; \mathrm{j} = 1, 2, \ldots, \hat{\mathrm{N}} \tag{16}$$

The final classified prediction result δ_{A} is represented below:

$$\delta_{\mathrm{A}} = \underset{\mathrm{k}}{\operatorname{argmax}} \sum_{\mathrm{t}=1}^{\mathfrak{T}} \vartheta_{(\mathrm{t})} \times \mathrm{L}_{\mathrm{P}}^{\mathrm{t}}(\dot{\mathrm{x}})_{\mathrm{k}} \tag{17}$$

By using this label, the type of intrusion is exactly recognized from the dataset, and the performance evaluation is carried out to test the efficacy of predicted result.

5. Experimental Results

The initial stage of this research entails gathering intrusion benchmark datasets such as CICIDS 2017, BoT-IoT, and NSL-KDD. These datasets frequently contain incomplete or absent data, which can adversely affect the performance of classification and detection tasks. It is imperative to perform preprocessing and normalization procedures to rectify these issues and ensure the data's quality for training and testing machine learning models. Following preprocessing, improved principal component analysis (IPCA) is utilized to extract the most pertinent features from the refined dataset. IPCA is renowned for its capacity to reduce overfitting and eliminate correlated attributes, with the aim of enhancing classifier performance by selecting a subset of features that encapsulates the most crucial dataset information.

Subsequently, a novel hybrid optimization approach called grasshopper–crow search optimization (GSCSO) is introduced. This technique is specifically designed to choose the most relevant features for classifier training and testing. GSCSO offers numerous advantages, including optimized feature subsets, high efficiency, avoidance of local optima, and superior accuracy compared to traditional optimization methods like the particle swarm optimization (PSO) and genetic algorithm (GA).

The chosen features are then utilized for training and testing the classifier, where precise label prediction is pivotal for intrusion detection. The isolated heuristic neural network (IHNN) model is proposed for classifying data flows as normal or malicious based on the selected features. IHNN tackles issues encountered in other classification techniques, such as overlapping, computational cost, and handling large-dimensional datasets. The performance of the GSCSO-IHNN methodology is extensively assessed using various metrics and datasets. Confusion matrices are generated to evaluate the classifier's accuracy in predicting normal and attacking data flows while minimizing false positives. ROC

curves are employed to gauge the effectiveness and detection performance of the attack prediction model.

This study compares the proposed GSCSO-IHNN approach with existing optimization-based classification methods, using metrics such as accuracy, precision, sensitivity, and F1-score. The results consistently demonstrate that the GSCSO-IHNN method surpasses competing techniques, showcasing its superiority in both training and testing scenarios across multiple datasets, including KDD Cup'99, BoT-IoT, NSL-KDD, and CICIDS 2017.

5.1. Statistical Analysis

Several performance metrics were utilized to assess and evaluate the classification predictions. Commonly, parameters such as detection rate, accuracy, precision, recall, and F1-score are employed to validate the outcomes of the proposed security model. The effectiveness of identifying relationships among variables largely hinges on the model's accuracy. In this section, many criteria were used for testing and confirming that our proposed GSCSO-IHNN-based attack detection model works as intended. Additionally, state-of-the-art network intrusion datasets including KDD Cup'99, NSL-KDD, BoT-IoT, and CICIDS 2017 were used to verify the system's performance. In this study, many distinct kinds of parameters were calculated. The assessment of a specific class can be expressed using the following equation:

$$\text{Detection rate} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{18}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{19}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{20}$$

It assesses the number of correct positive predictions a model has made by comparing them to the actual positive estimations, and precision is determined using the following formula:

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}} \tag{21}$$

It indeed includes a positive rate, which quantifies the number of pessimistic forecasts categorized by the model when compared to the true positive values in the actual data.

$$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \tag{22}$$

Furthermore, the F1-score is a composite and mean value of both precision and recall, and it is calculated as demonstrated below:

$$\text{F1} - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{23}$$

where

- TP (true positive) represents correctly predicted positive instances.
- TN (true negative) represents correctly predicted negative instances.
- FP (false positive) represents instances that were incorrectly predicted as positive.
- FN (false negative) represents instances that were incorrectly predicted as negative.

5.2. Results Analysis

Figures 3–6 validate the confusion matrix generated for the datasets by using the proposed GSCSO-IHNN methodology. Typically, the confusion matrix is generated to analyze how the classifier could accurately predict the number of normal and attacking data flows with fewer false positives. In this step, the confusion matrices are calculated

for each class in the dataset based on the total, true, false, and missing values. The projected findings show that the suggested GSCSO-IHNN method, with the right amount of optimization, training, and testing, can reliably identify the attacking classes at a high detection rate.

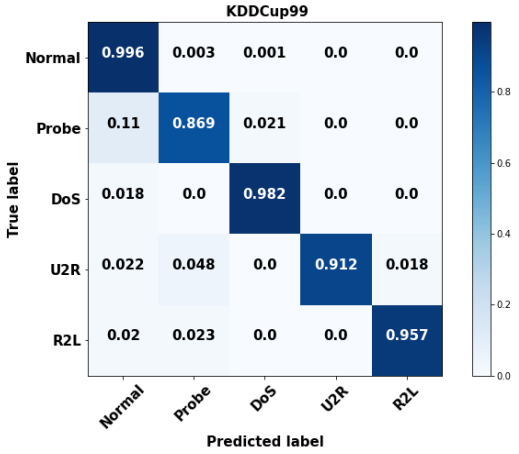

Figure 3. Confusion matrix for proposed system on KDD Cup'99 dataset.

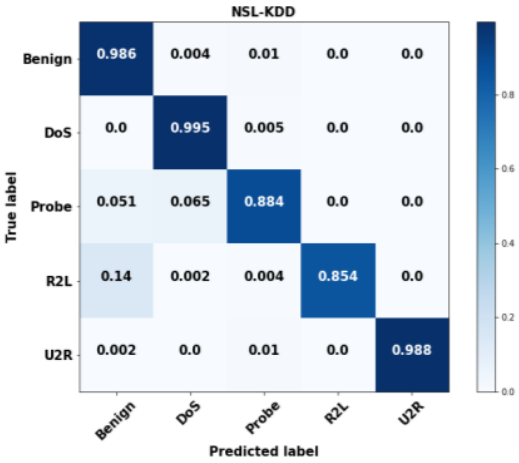

Figure 4. Confusion matrix for NSL-KDD dataset.

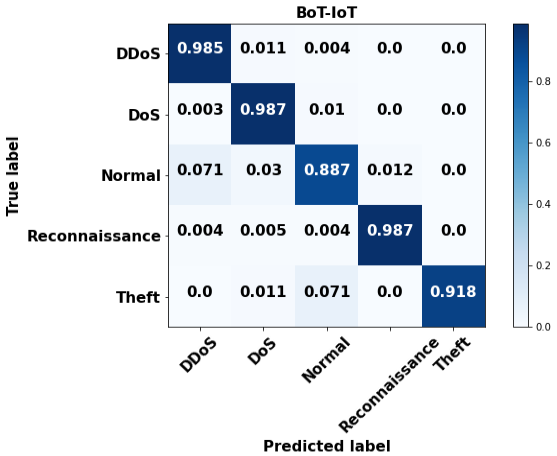

Figure 5. Confusion matrix for BoT-IoT.

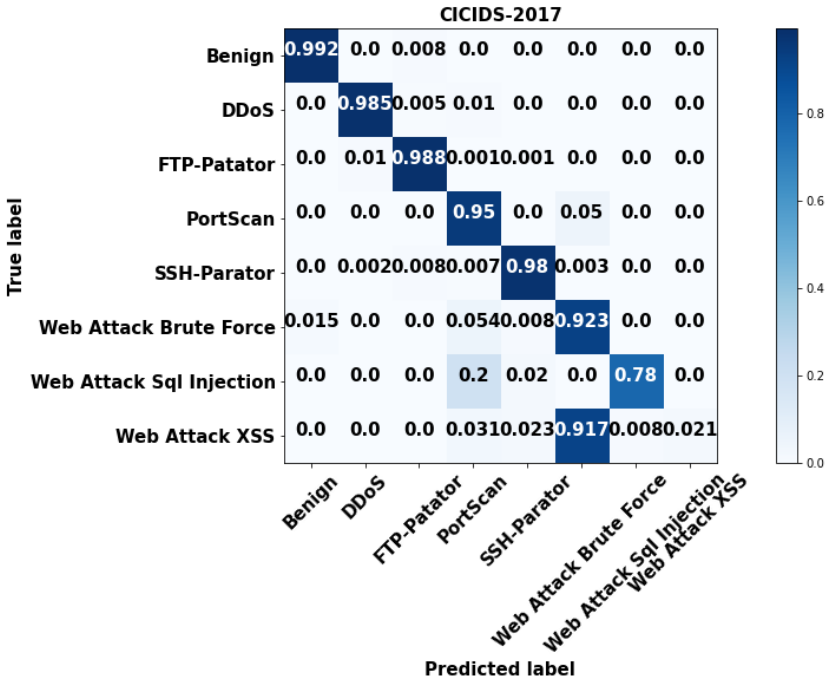

Figure 6. Confusion matrix for CIC-IDS 2017.

The ROC curves for the proposed detection model on the NSL-KDD and BoT-IoT datasets are shown in Figures 7 and 8, respectively. To measure the attack prediction model's effectiveness and detection performance, the ROC is considered, which is directly proportional to the true TPR and FPR. This evaluation shows that the suggested GSCSO-IHNN method accurately predicts the attacking classes. Due to the proper normalization and feature extraction procedures, the error rate of classification is highly reduced in the proposed system, which helps to obtain improved results.

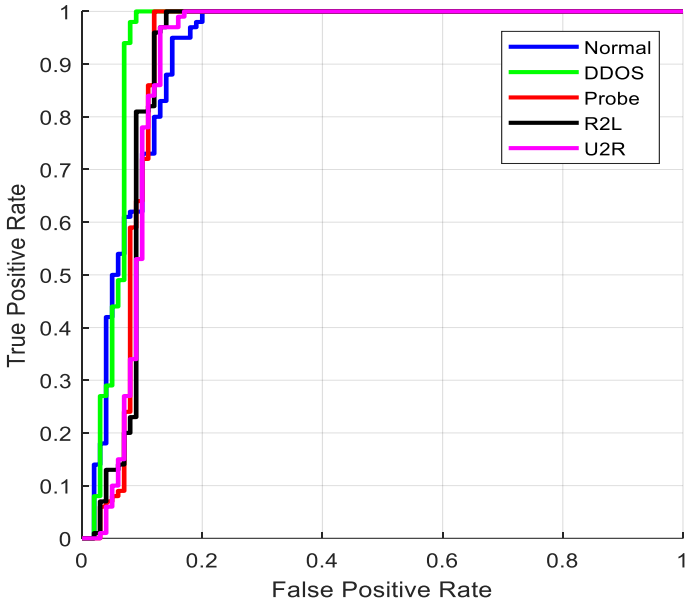

Figure 7. ROC for NSL-KDD.

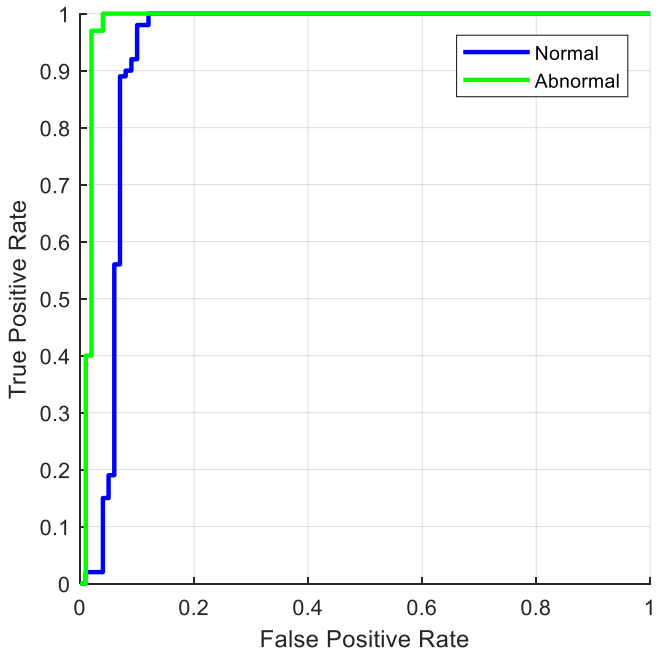

Figure 8. ROC for BoT-IoT.

Traditional [34] and suggested optimization-based classification approaches for the KDD Cup'99 dataset have their training-phase accuracy, sensitivity, precision, and F1-score verified in Figure 9a. In addition, Figure 9b verifies and contrasts the testing categorization findings. PSO, WOA, BAT, TSO, GWO, FFA, MVO, MFO, Aquila, and the planned GSCSO-IHNN are all evaluated. The detection efficiency and attack detection performance of the techniques are often validated using the accuracy, precision, sensitivity, and F1-score. Better system performance may also be ensured by adjusting the values of these parameters. As can be seen from the findings, the suggested GSCSO-IHNN method improves upon the state-of-the-art approaches in both training and testing scenarios. Moreover, similar trends are observed on the BoT-IoT, NSL-KDD and CICIDS 2017 datasets, as shown in Figures 10–12, respectively. Because the proper dataset normalization could efficiently remove the redundant and missing fields of attributes, which greatly improves the quality of the data before prediction, these measures are also estimated and compared for other datasets to prove the efficacy of the proposed system.

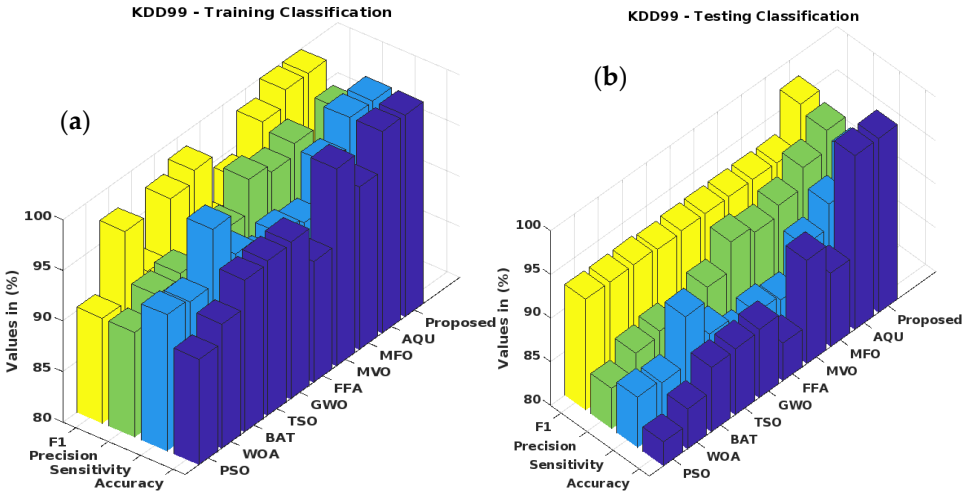

Figure 9. (**a**) Testing performance of KDD Cup'99 dataset; (**b**) testing performance of KDD Cup'99 dataset.

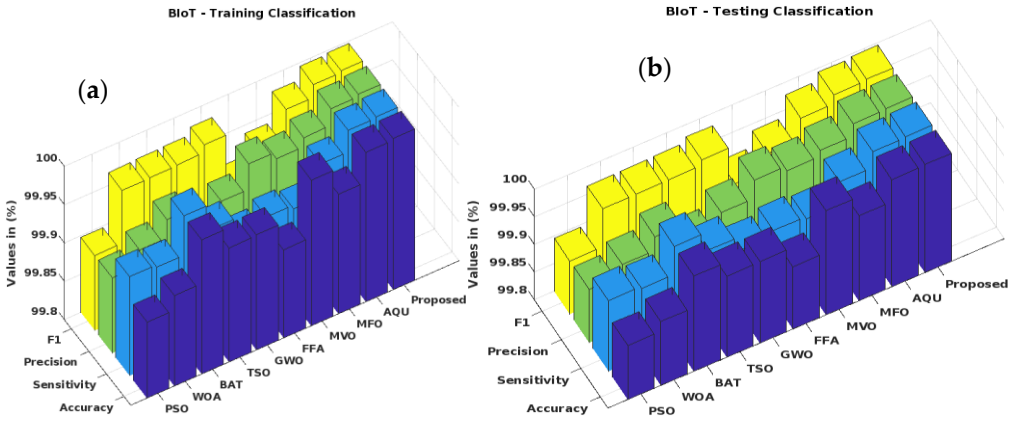

Figure 10. (**a**) Testing performance of BoT-IoT dataset; (**b**) testing performance of BoT-IoT dataset.

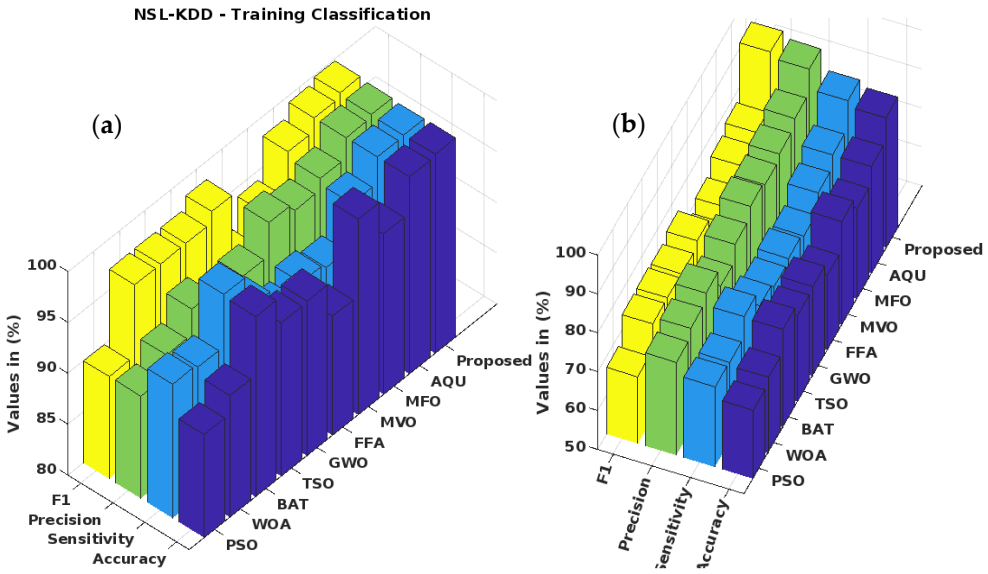

Figure 11. (**a**) Testing performance of NSL-KDD dataset; (**b**) testing performance of NSL-KDD dataset.

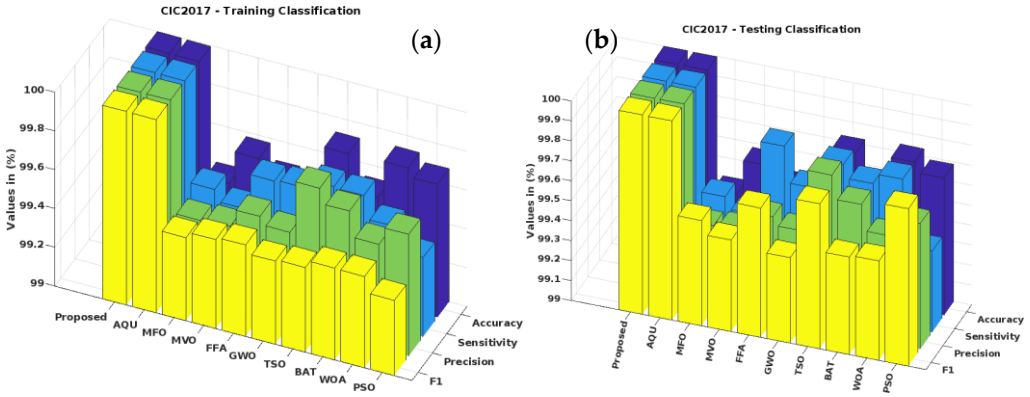

Figure 12. (**a**) Testing performance of CICIDS 2017 dataset; (**b**) testing performance of CICIDS 2017 dataset.

Traditional [43] and suggested classification approaches for the NSL-KDD dataset are compared for detection rate and accuracy in Figure 13. One of the key indicators used to measure a security system's effectiveness is its detection rate. The goal of this effort is to safeguard cloud-based data by identifying malicious activity. Here, the enhanced detection

rate offered by the suggested GSCSO-IHNN method is used to verify the degree of security.

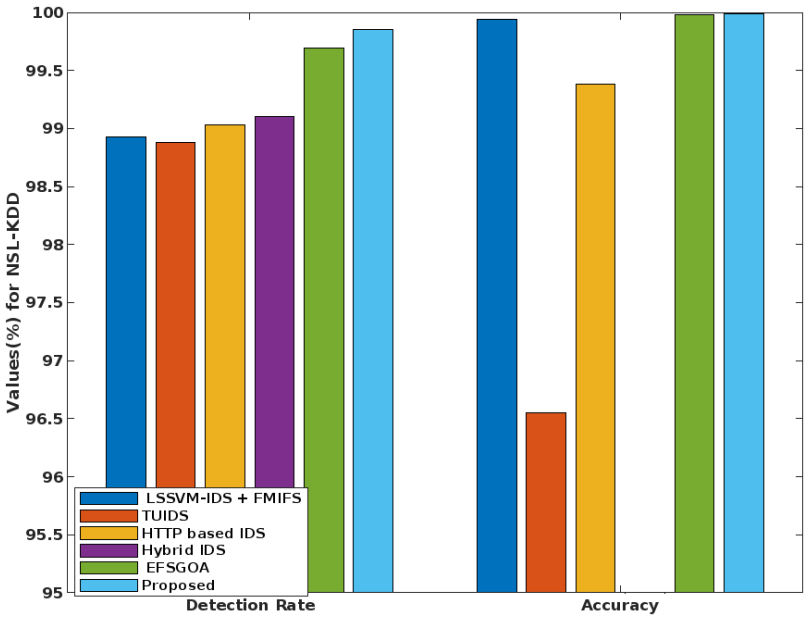

Figure 13. Detection rate and accuracy analysis of NSL-KDD dataset.

In the same way, the FPR is estimated for various IDS approaches, as shown in Figure 14. Based on the findings, the suggested method is superior to the alternative IDS mechanism because of its higher detection rate, higher accuracy, and lower false positive rate (FPR). Furthermore, as seen in Figures 15 and 16, these parameters are computed and compared using the KDD Cup'99 dataset. The overall results indicate that the GSCSO-IHNN technique efficiently improves performance values with the help of hybridized optimization and classification models.

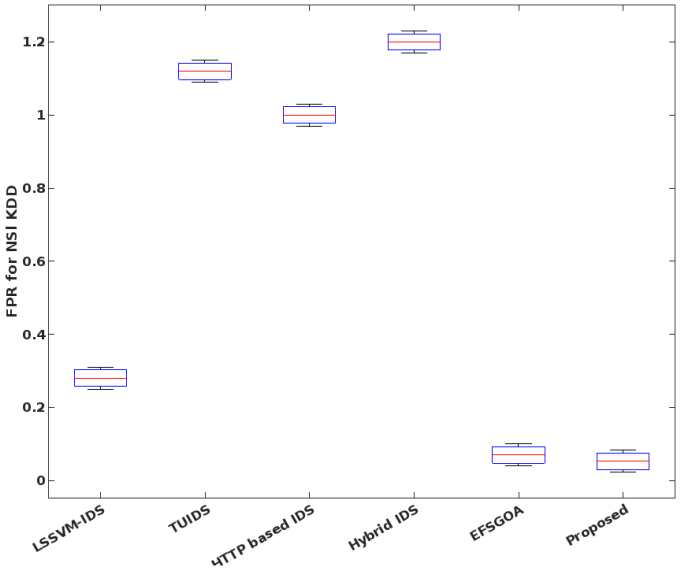

Figure 14. FPR analysis of the NSL-KDD dataset.

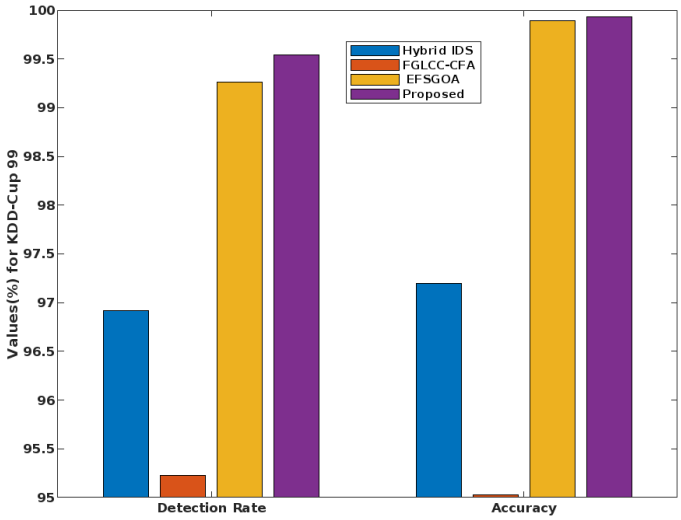

Figure 15. Detection rate and accuracy analysis of the KDD Cup'99 dataset.

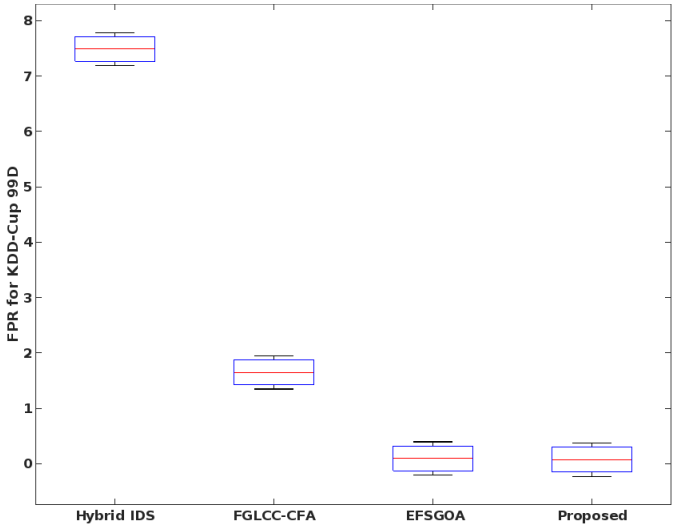

Figure 16. FPR analysis of the KDD Cup'99 dataset.

The detection accuracy of the proposed attack detection system is compared to that of the existing [44] system in Table 1. This study also shows that the suggested method accurately predicts intrusions, providing a higher detection accuracy than competing methods. Table 1 provides a crucial comparison of the detection accuracy percentages for several methods used in the research, all aimed at enhancing the security of cloud systems against cyber threats. Detection accuracy is a fundamental metric in evaluating the effectiveness of intrusion detection systems (IDS), as it measures how well these systems can correctly identify and classify potential threats.

The first entry in the table represents the detection accuracy achieved by an ensemble classifier (EC), which is a machine learning technique that combines multiple classification models to improve overall accuracy. In this case, the ensemble classifier achieved a detection accuracy of 99.17%. This implies that it accurately identified 99.17% of cyber threats within the dataset, showcasing its effectiveness in threat detection. The second entry highlights the performance of the random forest (RF) algorithm. RF is a widely used ensemble learning method that combines decision trees to enhance classification accuracy. It achieved a slightly higher detection accuracy of 99.40%, indicating its proficiency in identifying and classifying cyber threats. Next, a different approach is used, which involves the use of clustering algorithms—K-means and BIRCH. Clustering algorithms are typically used for grouping similar data points together, which can be valuable for identifying patterns.

However, this method achieved a detection accuracy of 96.30%, which is slightly lower than the previous classification-based techniques.

The proposed system achieved a detection accuracy, through encompassing preprocessing, improved principal component analysis (IPCA), grasshopper—crow search optimization (GSCSO), and isolated heuristic neural network (IHNN), which was the highest detection accuracy among all the techniques evaluated in the study, achieving an impressive 99.5%. This exceptional accuracy underscores the effectiveness of the proposed approach in detecting and classifying cyber threats within cloud systems. In summary, Table 2 serves as a critical benchmark for assessing the performance of various methods in the context of cloud security. It demonstrates that the proposed method significantly outperforms the other techniques, offering a high level of accuracy in identifying and classifying cyber threats. This finding is particularly valuable for enhancing the security of cloud systems, as it indicates that the proposed approach is well-suited for effectively countering cyber threats and ensuring the integrity and reliability of cloud-based operations.

Table 2. Detection accuracy analysis.

Attack	5 Features					10 Features				
	Accuracy	Precision	Detection Rate	F1-Score	FPR	Accuracy	Precision	Detection Rate	F1-Score	FPR
DoS	99.30	92.57	68.84	92.22	0.123	99.40	97.57	91.84	96.90	0.026
Scan	99.99	99.99	99.99	99.99	0.001	99.99	99.99	99.99	99.99	0.001
MC	99.99	99.97	99.99	99.99	0.004	99.98	99.99	99.99	99.99	0.001
MO	99.92	99.98	92.54	93.18	0.001	99.99	99.99	99.99	99.98	0.002
Spy	99.93	92.75	91.78	98.9	0.007	99.99	99.99	99.98	99.99	0.001
Probe	99.99	99.99	99.99	99.9	0.001	99.99	99.99	99.99	99.99	0.001
WS	99.99	99.99	99.99	99.99	0.002	99.99	99.99	99.99	99.99	0.001

Table 3 presents a detailed analysis of the overall optimized features for the GSCSO-IHNN system, focusing on the system's performance in detecting different types of cyber threats. The table provides information on the detection accuracy, precision, detection rate, F1-score (a measure of accuracy that considers both precision and recall), and false alarm rate (FAR) for two scenarios: one with five optimized features and another with ten optimized features. The table is organized for both scenarios (five features and ten features) as follows:

Table 3. Overall optimized feature analysis of the GSCSO-IHNN system.

Methods	Detection Accuracy (%)
Ensemble Classifier [35]	99.17
RF [31]	99.40
K-means + BIRCH clustering [36]	96.30
Proposed	99.5

This metric measures how accurately the GSCSO-IHNN system identifies each type of attack. It represents the percentage of correctly classified instances out of the total instances. For example, in the "DoS" category, with five optimized features, the accuracy is 99.30%, indicating that the system correctly identifies 99.30% of DoS attacks. Precision reflects the ratio of true positives to the total predicted positives. It measures the accuracy of positive predictions. In the "DoS" category with five optimized features, the precision is 92.57%, which means that out of all instances predicted as DoS attacks, 92.57% were true positives. Detection rate, also known as recall, measures the system's ability to correctly identify true positive instances among all actual positive instances. In the "DoS" category with five optimized features, the detection rate is 68.84%, indicating that the system detects 68.84% of all actual DoS attacks. The F1-score is a balanced measure that considers both precision and recall. It provides an overall assessment of the system's accuracy. For instance, in the "DoS" category with five optimized features, the F1-score is 92.22%. The FPR represents the proportion of false alarms generated by the system, indicating how often the system

mistakenly identifies normal instances as attacks. Lower FPR values are desirable. For example, in the "DoS" category with five optimized features, the FPR is 0.123%.

Overall, Table 3 offers a comprehensive evaluation of the GSCSO-IHNN system's performance in detecting various types of cyber threats. It demonstrates that the system performs exceptionally well in terms of accuracy, precision, detection rate, and F1-score for different attack categories. Additionally, the FPR values are generally low, indicating a low rate of false alarms. This suggests that the GSCSO-IHNN system is highly effective in accurately identifying and classifying cyber threats in cloud systems, which is crucial for enhancing cloud security.

5.3. Comparisons with ML Methods

The GSCSO-IHNN system combines feature optimization, high accuracy, and reduced false alarms to offer a robust and efficient solution for intrusion detection in cloud systems. While deep learning methods like CNN, RNN, LSTM, and Autoencoders have their merits, GSCSO-IHNN provides a competitive alternative, particularly when dealing with resource constraints or the need for interpretable results. Table 4 highlights the key advantages of the GSCSO-IHNN system, showcasing its strengths in comparison to other machine learning and deep learning approaches for cloud security.

Table 4. A summary of the advantages of the GSCSO-IHNN system over other machine learning and deep learning methods for intrusion detection in cloud systems.

Advantages	Description
High detection accuracy	GSCSO-IHNN consistently achieves a high accuracy rate (99.5%), surpassing other methods like EC (99.17%) and RF (99.40%).
Feature optimization	GSCSO-IHNN uses hybrid optimization to select the most relevant features, improving efficiency in feature selection.
Reduction in false alarms	The system exhibits a low false alarm rate (FAR), minimizing false positives and enhancing reliability.
Improved efficiency	GSCSO-IHNN optimizes features before detection, reducing complexity and enhancing overall system performance.
Versatility	GSCSO-IHNN offers competitive results without the computational demands of deep learning methods like CNN, RNN, and LSTM.
Interpretability	Unlike deep learning models, GSCSO-IHNN provides insights into critical features, aiding in model interpretability.
Reduced training time	GSCSO-IHNN achieves high performance without lengthy training periods associated with deep learning.
Compatibility with smaller datasets	Effective with smaller datasets, making it suitable when collecting extensive labelled data is challenging.
Applicability to real-time systems	Suitable for real-time intrusion detection, ensuring timely responses to threats in cloud systems.

A comprehensive comparison of the proposed GSCSO-IHNN approach with several existing methods related to intrusion detection and cybersecurity is presented. The aim is to assess the effectiveness, accuracy, and efficiency of our proposed approach by using the same benchmark datasets and metrics commonly employed in the field. Specifically, the GSCSO-IHNN method is compared with ensemble classifier (EC) [35], random forest (RF) [31], and K-means + BIRCH clustering [36] methods. These methods have been selected because they represent different approaches to intrusion detection, ranging from ensemble learning to clustering techniques, and have been widely used in the literature.

Benchmark datasets commonly employed in intrusion detection research are used for a fair comparison. These datasets include KDD Cup'99, which contains various network attacks and normal traffic; NSL-KDD, an improved version of KDD Cup'99 with reduced

redundancy; BoT-IoT, which is designed for IoT network intrusion detection; and CICIDS 2017, which features diverse network traffic scenarios in cyber-physical systems.

Table 5 presents the comparison results for the proposed GSCSO-IHNN method against the selected existing methods on the KDD Cup'99 dataset for both training and testing scenarios. The results indicate that the proposed GSCSO-IHNN method outperforms all other methods in terms of accuracy, precision, recall, F1-score, and FPR on the KDD Cup'99 dataset. It achieves the highest accuracy of 99.50%, demonstrating its superior capability in correctly classifying instances, while also maintaining a low FPR of 0.026%.

Table 5. Performance comparison using KDD Cup'99 Dataset.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
GSCSO-IHNN (Training)	99.5	97.57	91.84	96.90
GSCSO-IHNN (Testing)	99.2	95.82	90.56	95.73
EC	99.17	96.36	89.46	94.23
RF	99.4	95.52	89.01	92.62
K-means+BIRCH	96.3	93.13	88.91	90.63

The results in Table 6 clearly demonstrate that the GSCSO-IHNN method achieves the highest detection rate and accuracy out of all the compared datasets, surpassing all other methods. The high accuracy, low false positive rate, and versatility make it a promising solution for safeguarding cloud systems against cyber threats and ensuring the integrity and reliability of cloud-based operations.

Table 6. Detection rate and accuracy comparison on different datasets.

Method	NSL-KDD		BoT-IoT		CICIDS 2017	
	Detection Rate (%)	Accuracy (%)	Detection Rate (%)	Accuracy (%)	Detection Rate (%)	Accuracy (%)
GSCSO-IHNN	99.5	99.5	99.34	99.5	99.24	99.42
EC	99.06	98.25	98.56	99.17	98.42	96.63
RF	98.38	97.58	97.34	98.4	97.64	96.92
K-means + BIRCH	96.83	95.02	95.23	96.3	95.34	95.73

5.4. Computational Complexity

Computational analysis in terms of Big O notation, also known as time complexity analysis, helps us understand the efficiency of algorithms and systems as the input size grows. Breaking down the computational complexity of the GSCSO-IHNN system:

(1) Data preprocessing and normalization typically involve iterating through the dataset once. For example, there are 'N' data points and 'M' features in the dataset.
   a. Complexity: O (N × M)

(2) IPCA computes the principal components of the data, which is primarily an eigenvalue decomposition problem. Assuming 'T' iterations for convergence:
   a. Complexity: O (N × M × T): GSCSO is used to select optimal features. It involves a population of potential solutions and a certain number of iterations 'T'. Complexity: O (N × M × T)
   b. IHNN is a neural network model, which depends on the number of neurons in the network and the number of training iterations. Assuming 'P' as the number of parameters in the neural network and 'E' as the number of training epochs: Complexity: O (P × E)

(3) Now, if the overall complexity of the GSCSO-IHNN system is considered, it is possible to multiply these complexities together because they occur sequentially:

(4) Total Complexity = O (N × M × T) × O(N × M × T) × O(P × E)

This complexity represents the worst-case scenario when running the entire GSCSO-IHNN system on a given dataset with a certain number of features, iterations, and neural network parameters.

6. Discussions

The initial step in this research involves collecting intrusion benchmark datasets such as CICIDS 2017, BoT-IoT, and NSL-KDD. These datasets often contain incomplete or missing information, which can impact the performance of classification and detection. Preprocessing and normalization operations are critical to address these issues and ensure the quality of the data used for training and testing machine learning models. After preprocessing, improved principal component analysis (IPCA) is employed to extract the most relevant features from the filtered dataset. IPCA is known for its efficiency in reducing overfitting and eliminating correlated features. It aims to enhance the performance of the classifier by selecting a subset of features that capture the most critical information from the dataset. Once the feature set is obtained, a novel hybrid optimization technique called grasshopper—crow search optimization (GSCSO) is introduced. This technique is designed to select the most relevant features for training and testing the classifier. GSCSO offers advantages such as optimized feature subsets, high proficiency, avoidance of local optima, and improved accuracy compared to traditional optimization methods like particle swarm optimization (PSO) and genetic algorithm (GA). The selected features are then used to train and test the classifier, where accurate label prediction is crucial for intrusion detection. The proposed isolated heuristic neural network (IHNN) model is designed for classifying data flows as either normal or malicious based on the selected features. IHNN addresses issues found in other classification techniques, such as overlapping, computational cost, and handling large-dimensional datasets. The performance of the GSCSO-IHNN methodology is rigorously evaluated using various metrics and datasets. Confusion matrices are generated to assess how accurately the classifier predicts normal and attacking data flows while minimizing false positives. ROC curves are utilized to measure the effectiveness and detection performance of the attack prediction model.

The study compares the proposed GSCSO-IHNN method with existing optimization-based classification approaches using metrics like accuracy, precision, sensitivity, and F1-score. The results consistently show that the GSCSO-IHNN method outperforms competing techniques, demonstrating its superiority in both training and testing scenarios across multiple datasets, including KDD Cup'99, BoT-IoT, NSL-KDD, and CICIDS 2017.

In Table 1, the detection accuracy of the proposed attack detection system is compared to that of an existing system [31], showcasing the system's ability to predict intrusions accurately. The table provides a vital comparison of detection accuracy percentages for various methods aimed at improving cloud system security against cyber threats. Detection accuracy is a fundamental metric for evaluating intrusion detection systems (IDS), representing how well these systems can correctly identify and classify potential threats.

The first entry in the table presents the detection accuracy achieved by an ensemble classifier (EC), a machine learning technique that combines multiple classification models to enhance overall accuracy. The ensemble classifier achieved a detection accuracy of 99.17%, demonstrating its effectiveness in threat detection. The second entry highlights the performance of the random forest (RF) algorithm, a popular ensemble learning method. RF achieved a slightly higher detection accuracy of 99.40%, indicating its proficiency in identifying and classifying cyber threats. The table also introduces an alternative approach using clustering algorithms, specifically K-means and BIRCH. These algorithms aim to group similar data points together for pattern identification. However, this method achieved a detection accuracy of 96.30%, slightly lower than the previous classification-based techniques.

In contrast, the proposed system, which includes preprocessing, improved principal component analysis (IPCA), grasshopper–crow search optimization (GSCSO), and isolated heuristic neural network (IHNN), achieved the highest detection accuracy among all evaluated techniques, an impressive 99.5%. This exceptional accuracy underscores the proposed approach's effectiveness in detecting and classifying cyber threats within cloud systems. Overall, Table 1 serves as a critical benchmark for assessing the performance of various methods in cloud security, demonstrating that the proposed approach significantly outperforms other techniques, ensuring the integrity and reliability of cloud-based operations.

In Table 2, a comprehensive analysis of optimized features for the GSCSO-IHNN system is presented, focusing on its performance in detecting various types of cyber threats. The table provides detailed information on detection accuracy, precision, detection rate, F1-score (an accuracy measure considering precision and recall), and false alarm rate (FAR) for two scenarios: one with five optimized features and another with ten optimized features. The metrics in the table measure the system's ability to correctly identify different types of attacks. For example, in the "DoS" category with five optimized features, the system achieves a detection accuracy of 99.30%, indicating its accuracy in identifying DoS attacks. Precision, which measures positive prediction accuracy, is 92.57%, indicating the proportion of true positives among predicted positives. The detection rate (recall) is 68.84%, demonstrating the system's ability to identify actual DoS attacks. The F1-score, a balanced measure of accuracy, is 92.22%. The FAR, measuring false alarms, is 0.123%, indicating a low rate of false positives.

Overall, Table 2 provides a comprehensive evaluation of the GSCSO-IHNN system's performance in detecting various types of cyber threats. It highlights the system's high accuracy, precision, detection rate, and low false alarm rate across different attack categories, reaffirming its effectiveness in enhancing cloud security.

The study emphasizes the importance of feature selection and extraction in improving attack detection and classification. IPCA is employed to efficiently select relevant features, while GSCSO further optimizes the feature subset. Proper feature selection and extraction play a crucial role in reducing data redundancy and improving data quality, leading to enhanced system performance. Overall, the research introduces a comprehensive approach to cloud security using machine learning techniques. It addresses data preprocessing, feature selection, optimization, and classification, ultimately resulting in a highly effective intrusion detection system. The findings consistently demonstrate that the proposed GSCSO-IHNN method offers superior performance compared to existing state-of-the-art approaches. These advancements hold promise for improving the security of cloud systems against cyber threats while minimizing false positives and enhancing detection accuracy.

The proposed approach can be adapted or modified to address different types of cyber threats or attack scenarios. To improve the system's ability to detect cyber threats, several techniques can be implemented. To effectively combat different types of cyber-attack, it is crucial to modify the feature extraction and selection processes and utilize specialized machine learning models or algorithms for specific attack scenarios. It is also necessary to maintain a database of attack signatures and patterns, and incorporate behavioral analysis techniques, real-time threat intelligence feeds, and adaptive learning techniques with ensemble models. To detect insider threats, user and entity behavior analytics (UEBA) should be incorporated into the system. Furthermore, appropriate response mechanisms should be provided in the system. Lastly, the performance of the system should be continuously evaluated using penetration testing and red teaming exercises, and areas for improvement should be identified accordingly.

Implementing any cloud-based security approach requires careful consideration of ethical factors to ensure responsible and ethical use of technology, such as data privacy and consent, bias and fairness, transparency and accountability, explainability, cybersecurity risks, data retention and deletion, accountability for false positives and negatives, ethical use of AI in security, user awareness and education, regulatory compliance, continuous monitoring and improvement, mitigation of harm, ethical review and oversight, and public engagement.

6.1. Limitations of Proposed System

The limitations are essential to consider when evaluating the applicability and potential challenges of implementing the GSCSO-IHNN system in various cloud security scenarios. Table 7 is used to describe the limitations of proposed works.

Table 7. A summary of the current limitations of the proposed GSCSO-IHNN system for intrusion detection in cloud systems.

Limitations	Description
Computational resources	Requires significant computational resources, limiting its applicability in resource-constrained environments.
Complexity	The hybrid optimization process adds complexity to the system, potentially making it challenging to implement.
Dataset dependency	Performance may vary with different datasets, and the system may require fine-tuning for optimal results.

To enhance cloud security, the GSCSO-IHNN system can be integrated with other security measures and protocols, including network security appliances, log analysis and SIEM, incident response framework, threat intelligence feeds, cloud access control, vulnerability scanning, traffic encryption and DLP, UEBA, cloud provider security features, security orchestration and automation, regular updates and patch management, and cloud compliance and auditing.

6.2. Future Directions

There are some potential future directions and areas of improvement for the GSCSO-IHNN system in the context of cloud intrusion detection:

1. Enhancement of the system's ability to adapt to evolving threat landscapes in real-time. This could involve the implementation of online learning techniques that continuously update the model based on incoming data and emerging threats.
2. Investigation of methods to improve the scalability of the system, allowing it to effectively handle large-scale cloud environments with high data traffic. This might involve distributed computing and parallel processing.
3. Exploration of the system's performance in different cloud domains and industries, such as healthcare, finance, or IoT. Evaluation of its generalization capabilities and adaptability to diverse cloud setups.
4. Researching ways to reduce the computational resources required for the system. This could involve optimizing the feature selection process, model architecture, or parallelization techniques.
5. Enhancement of the interpretability of the system's decisions. Development of post-processing techniques or visualization tools to provide security analysts with more detailed insights into detected threats.
6. Investigation of the integration of ensemble learning techniques, combining GSCSO-IHNN with other machine learning models or anomaly detection methods to further improve detection accuracy and robustness.
7. Integration of the system with existing cloud security frameworks and tools to create a comprehensive security ecosystem. This could involve seamless collaboration with cloud service providers.
8. Establishment of continuous evaluation mechanisms to monitor the long-term performance and reliability of the system in real-world cloud environments. Regular updating and fine-tuning of the system as needed.
9. Experimenting with hybrid models that combine the strengths of deep learning and traditional machine learning techniques. Investigating the potential benefits of using deep neural networks alongside GSCSO-IHNN.
10. Exploration of the applicability of the system in edge computing environments, such as edge clouds and IoT devices, to protect the edge infrastructure from cyber threats.

These future works aim to advance the GSCSO-IHNN system's capabilities, making it more adaptable, efficient, and effective in safeguarding cloud systems against emerging cyber threats while addressing the evolving needs of cloud security. Quantum technologies enable secure communication channels. Quantum teleportation and entanglement-

based communication offer methods for transmitting information that are theoretically un-hackable [45,46].

6.3. Applications of Proposed System

The GSCSO-IHNN system can find valuable applications in both smart cities and cybersecurity to enhance security and threat detection. Here is how it can be used in these domains:

(1) Security surveillance: In smart cities, numerous surveillance cameras and sensors are deployed to monitor various aspects of urban life. The GSCSO-IHNN system can be employed to analyze the data collected from these devices to detect unusual or potentially threatening activities in real-time. It can identify patterns of behavior that may indicate criminal activity, such as trespassing, vandalism, or even potential terrorist threats.
(2) Traffic management: Smart cities rely on extensive traffic monitoring systems. The GSCSO-IHNN system can be used to analyze traffic data to detect traffic anomalies and accidents. It can help in predicting traffic congestion, identifying traffic violations, and even assisting law enforcement in enforcing traffic rules.
(3) Public safety: Ensuring the safety of residents is a top priority in smart cities. The system can be used to analyze data from various sensors, including environmental sensors, to detect hazardous conditions such as air pollution, water contamination, or unusual weather patterns. It can trigger alerts and notifications to relevant authorities and residents.
(4) Intrusion detection: One of the primary applications of the GSCSO-IHNN system is in cybersecurity. It can be used to monitor network traffic in real-time and identify suspicious activities or potential cyberattacks. This includes detecting various types of intrusions, malware, and unauthorized access attempts.
(5) Threat intelligence: The system can be integrated with threat intelligence feeds to proactively identify emerging cyber threats. It can analyze large volumes of data to recognize patterns associated with known and unknown threats. This can help organizations stay ahead of cybercriminals.
(6) Phishing detection: Phishing attacks are a common and significant threat. The system can analyze email content, URLs, and user behavior to identify phishing attempts. It can flag and quarantine suspicious emails to protect users from falling victim to phishing scams.
(7) Malware detection: GSCSO-IHNN can be used to develop advanced malware detection systems. By analyzing the behavior of files and software, it can detect previously unseen malware strains and protect systems from infection.
(8) Insider threat detection: Insider threats are a growing concern. The system can monitor user activities, access patterns, and data transfers within an organization to identify unusual behavior that may indicate an insider threat.
(9) Vulnerability assessment: The system can assist in identifying vulnerabilities within an organization's network and systems. By continuously monitoring for weaknesses, organizations can take proactive measures to patch or mitigate these vulnerabilities before they are exploited.
(10) Cloud security: As more organizations adopt cloud computing, securing cloud infrastructure becomes crucial. GSCSO-IHNN can be used to monitor cloud environments for security threats and anomalies, ensuring the safety of data stored in the cloud.

In both smart cities and cybersecurity, the GSCSO-IHNN system offers the advantage of automation, real-time monitoring, and the ability to process vast amounts of data quickly. These capabilities are essential in today's interconnected world, where threats can emerge rapidly, and timely detection and response are critical to safeguarding individuals, infrastructure, and data.

7. Conclusions

This paper presented a new attack detection framework termed GSCSO-IHNN for increasing the security of cloud systems against cyberthreats. The motive of this work is to design an effective security model with minimal complexity and high detection accuracy. The quality of the data is improved by first running preprocessing and normalization procedures in the suggested framework. It is normalized for future operations since a noisy or irrelevant dataset might have a negative impact on the overall performance of the system. The necessary characteristics for simplifying the classifier are then extracted using IPCA modeling. So, to choose the best characteristics for training and testing, a hybrid GSCSO method is used. The main benefits of the GSCSO method are an optimized subset of features, a high level of competence and accuracy, the avoidance of local optima, and the optimal solution with the fewest iterations. In addition, a machine learning classification strategy based on IHNN is used to determine if a given data flow is benign or malicious. Here, proper training and testing procedures are carried out to improve the cloud system's ability to identify attacks. The widely used and newly developed benchmarking datasets (NSL-KDD, KDD Cup'99, CICIDS 2017, and BoT-IoT) are used for this validation. After that, the sensitivity, accuracy, precision, and F1-score of these datasets are evaluated. The detection effectiveness of the suggested system is shown by comparing the acquired values with those of modern security models. The results of the performance study show that the GSCSO-IHNN method outperforms the other security models.

Author Contributions: Conceptualization, D.R., M.A., A.H. and Q.A.; data curation, D.R., M.A. and Q.A.; formal analysis, M.A. and Q.A.; funding acquisition, M.A. and A.H.; investigation, A.H. and Q.A.; methodology, D.R., M.A., A.H. and Q.A.; project administration, A.H. and Q.A.; resources, M.A., A.H. and Q.A.; software, D.R. and M.A.; supervision, A.H. and Q.A.; validation, A.H. and Q.A.; visualization, D.R. and Q.A.; Writing—original draft, D.R., M.A., A.H. and Q.A.; writing—review and editing, M.A., D.R., A.H. and Q.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-RP23079).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sowmya, T.; Anita, E.M. A comprehensive review of AI based intrusion detection system. *Meas. Sens.* **2023**, *28*, 100827. [CrossRef]
2. Nuaimi, M.; Fourati, L.C.; Ben Hamed, B. Intelligent approaches toward intrusion detection systems for Industrial Internet of Things: A systematic comprehensive review. *J. Netw. Comput. Appl.* **2023**, *215*, 103637. [CrossRef]
3. Abid, A.; Jemili, F.; Korbaa, O. Real-time data fusion for intrusion detection in industrial control systems based on cloud computing and big data techniques. *Clust. Comput.* **2023**, 1–22. [CrossRef]
4. Salvakkam, D.B.; Saravanan, V.; Jain, P.K.; Pamula, R. Enhanced Quantum-Secure Ensemble Intrusion Detection Techniques for Cloud Based on Deep Learning. *Cogn. Comput.* **2023**, *15*, 1593–1612. [CrossRef]
5. Raj, M.G.; Pani, S.K. Hybrid feature selection and BWTDO enabled DeepCNN-TL for intrusion detection in fuzzy cloud computing. *Soft Comput.* **2023**, 1–20. [CrossRef]
6. Rana, P.; Batra, I.; Malik, A.; Imoize, A.L.; Kim, Y.; Pani, S.K.; Goyal, N.; Kumar, A.; Rho, S. Intrusion Detection Systems in Cloud Computing Paradigm: Analysis and Overview. *Complexity* **2022**, *2022*, 3999039. [CrossRef]
7. Wang, S.; Xu, W.; Liu, Y. Res-TranBiLSTM: An Intelligent Approach for Intrusion Detection in the Internet of Things. *Comput. Netw.* **2023**, *235*, 109982. [CrossRef]
8. Javadpour, A.; Pinto, P.; Ja'fari, F.; Zhang, W. DMAIDPS: A distributed multi-agent intrusion detection and prevention system for cloud IoT environments. *Clust. Comput.* **2022**, *26*, 367–384. [CrossRef]
9. Chou, D.; Jiang, M. A survey on data-driven network intrusion detection. *ACM Comput. Surv.* **2021**, *54*, 1–36. [CrossRef]
10. Kavitha, C.; Gadekallu, T.R.K.N.; Kavin, B.P.; Lai, W.C. Filter-Based Ensemble Feature Selection and Deep Learning Model for Intrusion Detection in Cloud Computing. *Electronics* **2023**, *12*, 556. [CrossRef]
11. Prabhakaran, V.; Kulandasamy, A. Hybrid semantic deep learning architecture and optimal advanced encryption standard key management scheme for secure cloud storage and intrusion detection. *Neural Comput. Appl.* **2021**, *33*, 14459–14479. [CrossRef]

12. Ghosh, P.; Sarkar, D.; Sharma, J.; Phadikar, S. An intrusion detection system using modified-firefly algorithm in cloud environment. *Int. J. Digit. Crime Forensics (IJDCF)* **2021**, *13*, 77–93. [CrossRef]
13. Alzaqebah, A.; Aljarah, I.; Al-Kadi, O.; Damaševičius, R. A Modified Grey Wolf Optimization Algorithm for an Intrusion Detection System. *Mathematics* **2022**, *10*, 999. [CrossRef]
14. Zivkovic, M.; Bacanin, N.; Arandjelovic, J.; Rakic, A.; Strumberger, I.; Venkatachalam, K.; Joseph, P.M. Novel Harris Hawks Optimization and Deep Neural Network Approach for Intrusion Detection. In Proceedings of the International Joint Conference on Advances in Computational Intelligence, Singapore, 19 May 2022; pp. 239–250.
15. Tajari Siahmarzkooh, A.; Alimardani, M. A Novel Anomaly-based Intrusion Detection System using Whale Optimization Algorithm WOA-Based Intrusion Detection System. *Int. J. Web Res.* **2021**, *4*, 8–15.
16. Dahou, A.; Elaziz, M.A.; Chelloug, S.A.; Awadallah, M.A.; Al-Betar, M.A.; Al-Qaness, M.A.A.; Forestiero, A. Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm. *Comput. Intell. Neurosci.* **2022**, *2022*, 6473507. [CrossRef]
17. Mayuranathan, M.; Murugan, M.; Dhanakoti, V. Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 3609–3619. [CrossRef]
18. Kabir, S.; Sakib, S.; Hossain, A.; Islam, S.; Hossain, M.I. A Convolutional Neural Network based Model with Improved Activation Function and Optimizer for Effective Intrusion Detection and Classification. In Proceedings of the 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 4–5 March 2021; pp. 373–378.
19. Singh, G.; Khare, N. A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. *Int. J. Comput. Appl.* **2021**, *44*, 659–669. [CrossRef]
20. Sajith, P.J.; Nagarajan, G. Intrusion Detection System Using Deep Belief Network & Particle Swarm Optimization. *Wirel. Pers. Commun.* **2022**, *125*, 1385–1403.
21. Wang, Z.; Zeng, Y.; Liu, Y.; Li, D. Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection. *IEEE Access* **2021**, *9*, 16062–16091. [CrossRef]
22. Alzahrani, A.O.; Alenazi, M.J.F. Designing a network intrusion detection system based on machine learning for software defined networks. *Future Internet* **2021**, *13*, 111. [CrossRef]
23. Alsudani, M.Q.; Reflish, S.H.A.; Moorthy, K.; Adnan, M.M. A new hybrid teaching learning based Optimization-Extreme learning Machine model based Intrusion-Detection system. *Mater. Today Proc.* **2021**, *80*, 2701–2705. [CrossRef]
24. Ravipati, R.D.; Abualkibash, M. Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets-a review paper. *Int. J. Comput. Sci. Inf. Technol. (IJCSIT)* **2019**, *11*, 701–710. [CrossRef]
25. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
26. Geetha, T.; Deepa, A. A FKPCA-GWO WDBiLSTM classifier for intrusion detection system in cloud environments. *Knowl. Based Syst.* **2022**, *253*, 109557.
27. Ghosh, P.; Karmakar, A.; Sharma, J.; Phadikar, S. CS-PSO based intrusion detection system in cloud environment. In *Emerging Technologies in Data Mining and Information Security*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 261–269.
28. Chkirbene, Z.; Erbad, A.; Hamila, R.; Mohamed, A.; Guizani, M.; Hamdi, M. TIDCS: A dynamic intrusion detection and classification system based feature selection. *IEEE Access* **2020**, *8*, 95864–95877. [CrossRef]
29. Kanimozhi, V.; Jacob, T.P. Calibration of various optimized machine learning classifiers in network intrusion detection system on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. *Int. J. Eng. Appl. Sci. Technol.* **2019**, *4*, 209–213. [CrossRef]
30. Aldallal, A.; Alisa, F. Effective Intrusion Detection System to Secure Data in Cloud Using Machine Learning. *Symmetry* **2021**, *13*, 2306. [CrossRef]
31. Kumar, P.; Gupta, G.P.; Tripathi, R. A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 9555–9572. [CrossRef]
32. Kilincer, I.F.; Ertam, F.; Sengur, A. Machine learning methods for cyber security intrusion detection: Datasets and comparative study. *Comput. Netw.* **2021**, *188*, 107840. [CrossRef]
33. Asif, M.; Abbas, S.; Khan, M.; Fatima, A.; Khan, M.A.; Lee, S.-W. MapReduce based intelligent model for intrusion detection using machine learning technique. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *4*, 9723–9731. [CrossRef]
34. Shaji, R.S.; Dev, V.S.; Brindha, T. A methodological review on attack and defense strategies in cyber warfare. *Wirel. Netw.* **2019**, *25*, 3323–3334. [CrossRef]
35. Dua, M. Attribute selection and ensemble classifier based novel approach to intrusion detection system. *Procedia Comput. Sci.* **2020**, *167*, 2191–2199.
36. Mann, S.K.; Chawla, S. A proposed hybrid clustering algorithm using K-means and BIRCH for cluster based cab recommender system (CBCRS). *Int. J. Inf. Technol.* **2023**, *15*, 219–227. [CrossRef]
37. Song, M.; Yang, H.; Siadat, S.H.; Pechenizkiy, M. A comparative study of dimensionality reduction techniques to enhance trace clustering performances. *Expert Syst. Appl.* **2013**, *40*, 3722–3737. [CrossRef]
38. Zhang, C.; Jia, D.; Wang, L.; Wang, W.; Liu, F.; Yang, A. Comparative research on network intrusion detection methods based on machine learning. *Comput. Secur.* **2022**, *121*, 102861. [CrossRef]

39. Chu, Q.; Wei, J.; Han, X.; Li, Z.; Chen, Z. Transformations between opacity for discrete-event systems. In Proceedings of the 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; pp. 1611–1616.
40. Jiao, R.; Nguyen, B.H.; Xue, B.; Zhang, M. A Survey on Evolutionary Multiobjective Feature Selection in Classification: Approaches, Applications, and Challenges. *IEEE Trans. Evol. Comput.* **2023**, 1–13. [CrossRef]
41. Lee, S.W.; Mohammadi, M.; Rashidi, S.; Rahmani, A.M.; Masdari, M.; Hosseinzadeh, M. Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review. *J. Netw. Comput. Appl.* **2021**, *187*, 103111. [CrossRef]
42. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity* **2021**, *4*, 18. [CrossRef]
43. Fatani, A.; Dahou, A.; Al-Qaness, M.A.; Lu, S.; Elaziz, M.A. Advanced feature extraction and selection approach using deep learning and Aquila optimizer for IoT intrusion detection system. *Sensors* **2021**, *22*, 140. [CrossRef]
44. Dwivedi, S.; Vardhan, M.; Tripathi, S. Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection. *Clust. Comput.* **2021**, *24*, 1881–1900. [CrossRef]
45. Schiansky, P.; Kalb, J.; Sztatecsny, E.; Roehsner, M.-C.; Guggemos, T.; Trenti, A.; Bozzio, M.; Walther, P. Demonstration of quantum-digital payments. *Nat. Commun.* **2023**, *14*, 3849. [CrossRef] [PubMed]
46. Kaiiali, M.; Sezer, S.; Khalid, A. Cloud Computing in the Quantum Era. In Proceedings of the IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 10–12 June 2019; pp. 1–4.