Journal of Network and Systems Management (2023) 31:82
https://doi.org/10.1007/s10922-023-09767-8

Deep Learning Based Hybrid Intrusion Detection Systems to Protect Satellite Networks

Ahmad Taher Azar[1,2,3] · Esraa Shehab[4,5] · Ahmed M. Mattar[6] · Ibrahim A. Hameed[7] · Shaimaa Ahmed Elsaid[5]

Received: 17 May 2023 / Revised: 10 July 2023 / Accepted: 31 July 2023

Abstract
Despite the fact that satellite-terrestrial systems have advantages such as high throughput, low latency, and low energy consumption, as well as low exposure to physical threats and natural disasters and cost-effective global coverage, their integration exposes both of them to particular security challenges that can arise due to the migration of security challenges from one to another. Intrusion Detection Systems (IDS) can also be used to provide a high level of protection for modern network environments such as satellite-terrestrial integrated networks (STINs). To optimize the detection performance of malicious activities in network traffic, four hybrid intrusion detection systems for satellite-terrestrial communication systems (SAT-IDSs) are proposed in this paper. All the proposed systems exploit the sequential forward feature selection (SFS) method based on random forest (RF) to select important features from the dataset that increase relevance and reduce complexity and then combine them with a machine learning (ML) or deep learning (DL) model; Random Forest (RF), Long Short-Term memory (LSTM), Artificial Neural Networks (ANN), and Gated Recurrent Unit (GRU). Two datasets—STIN, which simulates satellite networks, and UNSW-NB15, which simulates terrestrial networks—were used to evaluate the performance of the proposed SAT-IDSs. The experimental results indicate that selecting significant and crucial features produced by RF-SFS vastly improves detection accuracy and computational efficiency. In the first dataset (STIN), the proposed hybrid ML system SFS-RF achieved an accuracy of 90.5% after using 10 selected features, compared to 85.41% when using the whole dataset. Furthermore, the RF-SFS-GRU model achieved the highest performance of the three proposed hybrid DL-based SAT-IDS with an accuracy of 87% after using 10 selected features, compared to 79% when using the entire dataset. In the second dataset (UNSW-NB15), the proposed hybrid ML system SFS-RF achieved an accuracy of 78.52% after using 10 selected features, compared to 75.4% when using the whole dataset. The model with the highest accuracy of the three proposed hybrid

Extended author information available on the last page of the article

Published online: 04 September 2023

DL-based SAT-IDS was the RF-SFS-GRU model. It achieved an accuracy of 79% after using 10 selected features, compared to 74% when using the whole dataset.

Keywords Satellite-terrestrial integrated network · STIN & UNSW-NB15 dataset · Intrusion detection system (IDS) · Cyber security · Machine learning · Deep learning

1 Introduction

A satellite is a self-contained communications device with the capacity to receive messages from Earth and retransmit them using a transponder, which functions as both a radio transmitter and receiver. Thanks to the satellite, large areas of Earth may be seen at once. As a result, satellites can gather data more quickly and efficiently than devices on the ground, as shown in Fig. 1. The ability of satellites to transmit signals from one place to numerous destinations is their fundamental benefit. As a result, "point-to-multipoint" communications like broadcasting are perfect for satellite technology [1]. Satellite communication is the best option for underserved and remote locations with dispersed populations because it doesn't require significant investments on the ground. However, the satellite-terrestrial network has significant issues with networks that are growing daily, enormous devices, the inclusion of the satellite network, data security, and data privacy [2]. Another issue is

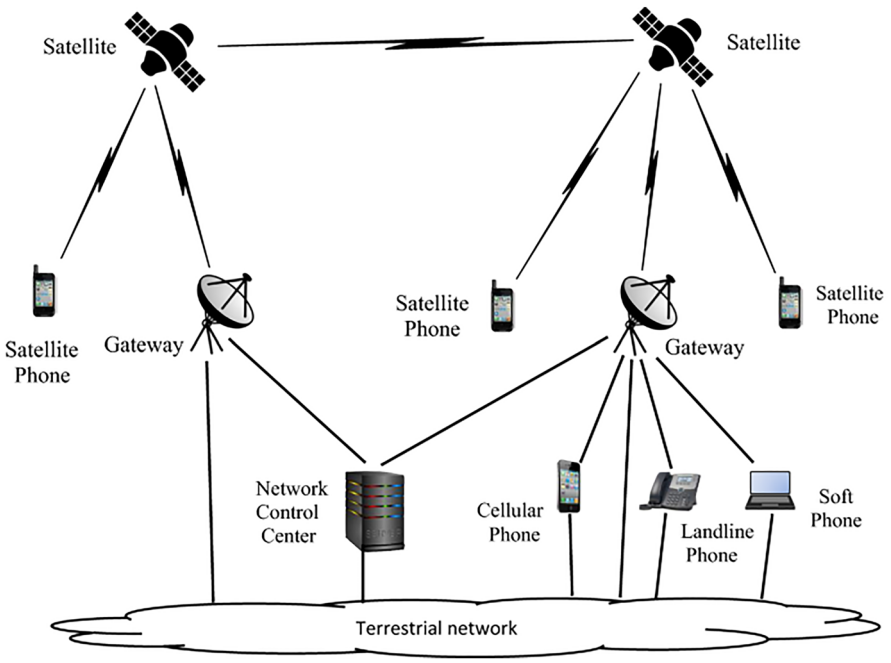

Fig. 1 Satellite-terrestrial communication networks [3]

that the satellite network has less computing power and resources than the terrestrial network. A satellite node that has been targeted by an attacker becomes quickly exhausted and is challenging to repair. Therefore, high-level protection for modern networks requires the development of effective intrusion detection techniques. To address these issues, further classification algorithms are presented. Therefore, high-level protection for modern networks necessitates the development of effective intrusion detection techniques due to the rise in network intrusion attacks [3].

There is a greater demand for network security due to the rise in network intrusion attacks [3, 4]. Cybersecurity is playing a crucial role in satellite system protection through the application of security policies, confidentiality, and integrity. Intrusion detection systems (IDSs) are utilized to detect attacks and any abnormal behavior in satellite systems [5]. IDS is categorized into five categories, as shown in Fig. 2. IDSs can be divided according to their structure: Based on centralized IDS or distributed IDS. The centralized IDS analyses data at a set number of locations [10]. The distributed IDS, which includes multiple IDS through a large network, all of which are connected to each other or connected to a central server that provides advanced network monitoring and incident analysis, can also be divided according to their deployment location into host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS) [8]. HIDS in the IDS system uses the activities of the system in the form of different log files running on the local host computer to detect attacks, but NIDS in the IDS system uses network behavior. The network behaviors are gathered using network equipment through mirroring by networking devices, such as routers, switches, and network taps, and analyzed to specify attacks and possible threats hidden within network traffic. The log files in NIDS are gathered through local sensors. While HIDS depends on the information in log files, which include system logs, sensor logs, file systems, software logs, user account information, disc resources, and others for each system, NIDS inspects the contents of each packet in network traffic flows. Many organizations utilize a hybrid of both HIDS and NIDS [8].

In addition, IDS can be divided according to the approach that is used to detect attacks and other hidden potential threats within network data [3–8] into two categories: Anomaly-Based detection and Signature-Based detection, also known as "misuse detection" or "knowledge-Based detection". Anomaly-Based detection has the feature of detecting deviations from normal behavior. The role of this technique is to: establish a baseline for the normal behavior of network traffic, then compare the incoming traffic with this baseline to detect malicious attacks. This IDS type enables detecting unknown attacks as well as known attacks [9]. Signature-Based detection has predefined signatures for known attacks that are matched with all connection patterns in the network to detect and stop any anomalous attacks. The main advantage of this type of IDS is that it detects known attacks. However, the unknown attacks are not detected due to the unavailability of attack signatures [9].

According to their response, IDSs are classified into passive IDS, which monitor, log, and provide alerts to activity, and active IDS, which act based on software design [10].

Based on the above figure, the proposed ML/DL-based hybrid IDSs in this research study focused on signature-based IDSs. It provides a high level of security

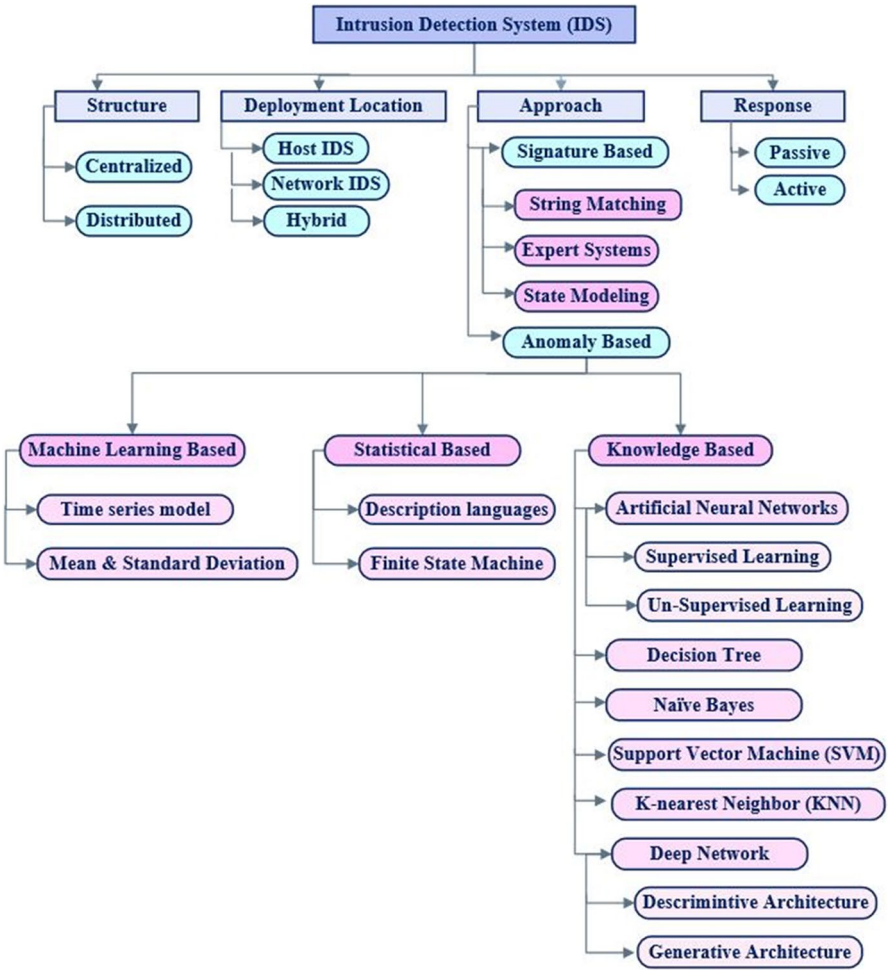

Fig. 2 Categories of intrusion detection systems (IDS) [10]

for both satellite and terrestrial networks by detecting known attacks and improving the accuracy of the results.

1.1 Problem Statements

In terms of bandwidth, computing power, and other resources, satellite networks and terrestrial networks differ significantly from one another. Furthermore, when satellites are launched, improving their hardware becomes much more challenging. The fundamental issue with satellites is that they have limited resources and are challenging to repair when attacked. Thus, Network Intrusion Detection Systems (NIDS) were developed to protect the modern network environment. NIDS is used to distinguish between the network's normal and anomalous traffic [2].

The effectiveness of NIDS is evaluated based on their performance in identifying attacks, which requires a comprehensive dataset that contains normal and abnormal behaviors, such as UNSW-NB15. However, this dataset and NIDS are generally used to simulate terrestrial networks and are difficult to apply to satellite networks because of the characteristics of STIN. Rather, the STIN dataset was used to simulate satellite networks.

1.2 Contribution

Our contribution is to propose four hybrid intrusion detection approaches for satellite-terrestrial systems in modern network environments by considering the data from satellite and terrestrial networks to achieve good detection performance of malicious activities in the network traffic. The key contribution of this paper is the proposal of four SAT-IDSs:

- The first model uses the sequential forward selection (SFS) method based on the Random Forest model (RF-SFS) to select important features in the dataset and then combine it with Random Forest (RF).
- The second model uses the sequential forward selection (SFS) method based on the Random Forest model (RF-SFS) to select important features in the dataset and then combine it with the Artificial Neural Network (ANN) model (RF-SFS-ANN) to achieve higher accuracy than ANN.
- The third model combines the RF-SFS with Long-Short-Term memory (RF-SFS-LSTM) to improve the accuracy of the LSTM model.
- The fourth one utilizes the RF-SFS with the Gated Recurrent Unit (RF-SFS-GRU).

This paper is constructed as follows: Sect. 2 presents the literature review. Section 3 describes the methodology. In Sect. 4, four novel SAT-IDSs are proposed. The results of the experiments are presented in Sect. 5 and discussed in Sect. 6. Section 7 sums up the paper and presents the future work.

2 Related Work

This section focuses on some recent related works on intrusion detection systems using UNSW-NB15 for terrestrial datasets and/or STIN for satellite datasets. Researchers in [11] provide a comprehensive review of the latest feature selection (FS) methods in large data sets, organized based on their nature, like search strategy, evaluation process, and feature structure.

The authors in [5] proposed a new approach-based intrusion detection method using data from satellite and terrestrial networks. The model combines random forest (RF) and multilayer perceptron (MLP) to increase the accuracy of intrusion detection compared to other machine learning models. They also analyze the efficiency of the proposed framework for the satellite and then use three datasets for

experiments, namely NSL-KDD, KDD-CUP 99, and STIN. Other researchers [12] analyzed the important features of huge traffic in networks to improve the accuracy of the intrusion detection model and minimize the execution time. They use the Information Gain method as a feature selection method to select important features and then implement Bayes Net (BN), Random Forest (RF), Naive Bayes (NB), J48, and Random Tree (RT) classifiers. The results of experiments on the CICIDS-2017 dataset improved the accuracy and execution time, and the Random Forest model (RF) achieved the highest accuracy of 99.86% based on 22 selected features, while the J48 model achieved an accuracy of 99.87% based on 52 selected features but with a longer execution time. The authors in [13] proposed a comparative study on Artificial Intrusion Detection Systems (AIDS) using different datasets and different types of attacks. They also apply different supervised ML algorithms such as random forest (RF), decision tree (DT), k-nearest neighbour (k-NN), support vector machine (SVM), naive Bayes (NB), artificial neural network (ANN), and convolutional neural network (CNN) algorithms, as well as unsupervised ML algorithms including k-means, expectation–maximization (EM), and self-organizing maps (SOM) algorithms. In [8], the authors propose an alert system based on hybrid intrusion detection through a highly scalable framework. The proposed framework handles large-scale data in real time.

The authors in [14] proposed a UAV and satellite-based 5G network security model based on machine learning to enhance the security of networks by effectively detecting vulnerabilities and cyberattacks. This approach is divided into two parts: the creation of the model using different machine learning algorithms and the implementation of the ML-based model using satellite or terrestrial gateways. This model achieves maximum accuracy with a 99.99% true negative rate and a 0% false negative rate by using a decision tree algorithm rather than other ML classifiers. Other researchers [15] proposed deep learning model-based intrusion detection using hybrid sampling to solve the problem of data imbalance. To produce a balanced dataset, they use one-sided selection (OSS) to reduce the majority samples and increase the minority samples using the Synthetic Minority Over-Sampling Technique (SMOTE). Then they implement Bi-directional long short-term memory (BiLSTM) using the NSL-KDD and UNSW-NB15 datasets, and they achieve 76.82% for multiclass classification and 77.16% for binary classification. In [16], authors proposed a new approach-based intrusion detection and compared it with different machine learning techniques, including SVM, AdaBoost, decision tree, and MLP, to classify normal and anomalous traffic. It depends on selected features based on the correlation between the features, and it implements using the UNSW-NB 15 dataset for network anomaly detection. The proposed approach achieves high accuracy in binary classification using Adaboost, which is 99.3%.

Researchers in [17] proposed an overview of different feature selection methods to reduce the computation time and improve the accuracy of the prediction. They focus on explaining the filter, wrapper, and embedded methods with various machine learning classifiers. Authors in [18] implemented a filter-based XGBoost algorithm using UNSW-NB15 to reduce feature space and then used ML approaches such as k-Nearest-Neighbour (kNN), Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), and Artificial Neural Network (ANN). The

results based on XGBoost-based feature selection achieve high test accuracy for the DT model, ranging from 88.13% to 90.85% in binary classification. The authors in [19] applied a Gated Recurrent Units (GRU) classifier based on deep learning to detect DDoS and intrusion attacks in a software-defined network (SDN). Other researchers in [20] studied the effect of different feature selection techniques such as Chi-Square, Information Gain (IG), and Recursive Feature Elimination (RFE) on the performance of ML classifiers, namely Naïve Bayes, Decision Tree Classifier, Support Vector Machine, k-nearest neighbours, Random Forest Classifier, Logistic Regression, and Artificial Neural Networks. Authors in [21] proposed a new hybrid approach using chi-square as a feature selection technique, Extra Tree, and ANOVA with ML classifiers such as Random Forest, k-Nearest Neighbours, Decision Tree, and XGBoost to detect DDoS attacks on IoT devices.

The authors in [22] presented an automatic evaluation method based on human–computer interaction and virtual reality for mental health physical examination. They use the HCI technique to induce the participants' emotional experience via three virtual reality (VR) emotional scenes: sad, joyful, and calm, and construct two differential pupil-waves for sad and joyful with the calm pupil-wave as the baseline. Researchers in [23] developed ECMADE, which is proposed to solve premature convergence and search stagnation. When compared to other algorithms, ECMADE outperforms them in terms of solution quality, space distribution, and robustness. Other researchers in [24] presented a multi-strategy competitive-cooperative co-evolutionary algorithm based on adaptive random competition and neighbourhood crossover, namely MSCOEA, to effectively solve multi-objective optimization problems (MOPs) and fully balance uniformity and convergence. The authors in [25] proposed a long short-term memory (LSTM)-based approach to detect network attacks using an SDN-supported intrusion detection system in IoT networks. They effectively identify the attacks and classify the attack types with an accuracy of 0.971. Researchers [26] proposed an approach that combined both CNN and GRU to optimize the network parameters. In this simulation, the authors used the CIC-IDS-2017 benchmark dataset and metrics such as precision, recall, false-positive rate (FPR), and true-positive rate (TPR). The authors also performed a comparative analysis with other existing approaches, and the obtained results indicate the efficacy of the proposed IDS scheme in real-world cybersecurity setups. Others [27] developed a deep learning model for network intrusion detection in both binary and multiclass classifications of network traffic. They also proposed an effective refinement strategy and generated several models for lowering the FNR and increasing predictability for the minority classes. Researchers in [28] proposed LSTM (Long Short-Term Memory) combined with FCN (Fully Connected Network) deep learning approaches to classify the normal and anomalous connections on intrusion datasets and specify the attack pattern more accurately. The proposed deep learning model achieved a better classification accuracy result using the KDDCup99, NSLKDD, GureKDD, KDDCorrected, Kyoto, and NITRIDS datasets, respectively. The authors in [29] proposed an unsupervised ensemble autoencoder connected to the Gaussian mixture model (GMM) to adapt to multiple domains. The performance of the proposed model is comparable with the selected anomaly detection baselines using three public datasets.

In recent related works on IDS, previous authors proposed many approaches to improving the accuracy results in binary and multi-class classification by combining two techniques with each other or solving an imbalanced dataset. As a result, some researchers proposed filter methods such as Chi-square and Information Gain, which were combined with ML models to improve accuracy. Other researchers proposed solutions to the imbalanced dataset problem, such as the SMOTE technique, which also increases accuracy. Their findings included some problems: mainly the low accuracy of the adapted models combined with more execution time and complexity time according to the number of used features. Many researchers used terrestrial network traffic as a dataset to evaluate those proposed approaches, as shown in Table 1.

3 Methodology

This section explains in detail the two datasets for satellite and terrestrial networks, STIN and UNSW-NB15, respectively. Then, the ML and DL models, which are implemented in the proposed SAT-IDS approaches to detect different satellite-terrestrial attacks, were reviewed.

3.1 Datasets

This section has two parts based on types of datasets, mainly STIN and UNSW-NB15, respectively.

3.1.1 STIN Dataset

The STIN dataset [2] was utilized to represent a satellite dataset that includes various types of attacks in modern satellite and terrestrial network environments. The authors in [2] simulate a real scenario for the terrestrial network and satellite network. This dataset contains two types of traffic: TER20 and SAT20, in csv format. These two files contain 32 features with labels and nine different types of attacks. The distribution of the samples for types of attacks in the training set for each category is mainly one for terrestrial network attacks, which include 7 various types of attacks like Botnet with 14,622 records, Web Attack with 13,017 records, Backdoor with 12,762 records, LDAP_DDos with 15,694 records, MSSQL_DDos with 15,688 records, NetBIOS_DDos with 11,530 records, and the last type is Portmap_DDos with 14,380 records. Another one is for satellite network attacks, which include two types of attacks: Syn_DDos with 54,789 records and UDP_DDos with 57,082 records.

3.1.2 UNSW-NB15 Dataset

The UNSW-NB15 dataset [30] was used to represent terrestrial traffic only, which includes various types of attacks in modern terrestrial network environments. This dataset contains two main files, mainly training set and testing set files in csv format,

Table 1 Summary of the latest research

Ref. (Year)	ML/DL Methods	Terrestrial/Satellite Datasets	Contribution	Limitations
2022 [5]	Integrates random forest (RF) and multilayer perceptron (MLP) to produce RFMLP model	KDD CUP 99, NSL-KDD, and STIN	Increasing intrusion detection performance	Low performance of the 'Syn_DDos' class in the STIN dataset
2020 [12]	Information Gain method-based Bayes Net (BN), Random Forest (RF), Naive Bayes (NB), J48, and Random Tree (RT) classifiers	CICIDS-2017	Improving the accuracy of the intrusion detection model and minimize the execution time	The proposed approach addressed a binary classification problem to detect attacks in the network traffic, regardless of the attack category
2020 [15]	BiLSTM model-based Synthetic Minority Over-Sampling Technique (SMOTE)	NSL-KDD and UNSW-NB15	Solve the problem of data imbalance using one-sided selection (OSS) to reduce the majority samples and increase the minority samples	Low accuracy results
2022 [16]	SVM, AdaBoost, decision tree, and MLP depends on selected features based on the correlation between the features	UNSW-NB 15	A new approach-based intrusion detection to classify normal and anomalous traffic	The proposed approach addressed a binary classification problem to detect attacks in the network traffic, regardless of the attack category
2020 [18]	Filter based XGBoost algorithm	UNSW-NB15	Implemented a filter-based XGBoost algorithm using UNSW-NB15 to reduce feature space and then used ML approaches such as k-Nearest-Neighbour (kNN), Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), and Artificial Neural Network (ANN)	Low accuracy results

and it contains 45 features with labels and nine different types of attacks. The distribution of the samples for types of attacks in the training set is Analysis with 677 records; Backdoor with 583 records; DoS with 4089 records; Exploits with 11,132 records; Fuzzers with 6,062 records; Generic with 18,871 records; Reconnaissance with 3,496 records; Shellcode with 378 records; and Worms with 44 records. However, the distribution of the samples for types of attacks in the testing set for UNSW-NB15 is: Backdoor with 1,746 records; Analysis with 2,000 records; DoS with 12,264 records; Exploits with 33,393 records; Fuzzers with 18,184 records; Generic with 40,000 records; Worms with 130 records; Shellcode with 1,133 records; and Reconnaissance with 10,491 records. Table 2 compares the two datasets used, the STIN and UNSW-NB15 datasets.

3.2 ML & DL techniques

Although most researchers propose IDSs based on machine learning (ML), deep learning (DL) models are less commonly used in cybersecurity since they require enormous amounts of high-quality data for appropriate training. DL is a subset of machine learning that implements artificial neural networks [31]. It has the structure of a multi-layer neural network. Traditional ML algorithms perform better when data volumes are small, and training takes little time. On the contrary, the test time of the ML takes more time based on the amount of data. This leads to the conclusion that using DL algorithms achieves higher accuracy than ML when increasing the volume of data and consuming less test execution time. This article investigates both ML-and DL-based intrusion detection systems (IDSs) as efficient solutions for detecting network intrusions. This research work considers RF as an example of ML and Long-Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Artificial Neural Networks (ANN) as DL techniques. The following subsections provide a review of the supervised ML and DL approaches that are used in this work:

3.2.1 Random Forest (RF)

Random forest (RF) [32] is a machine learning method dependent on multiple decision trees that can be used for tasks in regression and classification. The difference

Table 2 Comparison between UNSW-NB15 and STIN datasets

	STIN	UNSW-NB15
Domain	Satellite Network Traffics	Terrestrial Network Traffics
Total no. of attack types	2 satellite-type attacks and 9 terrestrial-type attacks	9 types of attacks
No. of features	32 features	45 features
Total no. of attacks records in train set	209,564 records	45,332 records
Total no. of normal records in train set	No records	37,000 records
Total no. of attacks records in test set	41,913 records	119,341 records
Total no. of normal records in test set	No records	56,000 records

between a random forest and a decision tree is to avoid overfitting for better generalizability by adding randomness for multiple decision trees in a random forest. The results of the prediction for a random forest after training have a higher probability through a voting method based on the prediction of each tree [20]. RF has an advantage that increases the performance of the model by producing an important score for each feature and removing features with the lowest score [32].

3.2.2 Artificial Neural Network (ANN)

A feed-forward artificial neural network with many hidden layers is called a multilayer perceptron (MLP). The number of neurons in the MLP output layer for classification issues is equal to the number of classes to be identified, but the number of neurons in the input layer is associated with the number of features. Backpropagation is frequently used to train the layers between the input and output layers [33]. These layers are completely connected. Equation 1 demonstrates how the network calculates the output of each layer during forward propagation based on an activation function from the prior layer as well as related weight and bias variables, as shown in Eq. 1 [14].

$$Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}, \tag{1}$$

where l stands for input layers, $\boldsymbol{W^{[l]}}$ for the weight matrix, $\boldsymbol{b^{[l]}}$ for the bias vector, and $\boldsymbol{Z^{[l]}}$ for the output matrix.

An activation function can be used to normalize the output of an MLP because the output could have any value. Equation 2 illustrates how the activation function can change each layer's output to fall inside a specific range [32].

$$A^{[l]} = g\left(Z^{[l]}\right), \tag{2}$$

where $A^{[l]}$ denotes the output matrix that has been activated, and $g(Z^{[l]})$ is the type of activation function that is used in the hidden layers. The suggested methodologies used activation functions such as "tanh" and "softmax" for the hidden layer and final output layer, respectively. A straightforward activation function called tanh, which is defined in Eq. 3 [32], provided higher performance for multi-layer neural networks. The "softmax" activation function described in Eq. 4 [32] is typically used for multi-classification because it can correct sigmoid function flaws and guarantee that the probability sum of the output layer is equal to 1. A cross-entropy loss function is used to calculate the error between the predicted value and the actual value.

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}, \tag{3}$$

$$a_{\text{soft}} = \frac{e^{z_i}}{\sum_{j=1}^{j} e^{z_j}} \quad for\, i \in [1, j], \tag{4}$$

where a_{soft} in Eq. 4 denotes to "softmax" activation function, $\boldsymbol{J}$ is the class number, $\boldsymbol{z_i}$ represents the $\boldsymbol{ith}$ output value.

3.2.3 Long-Short Term Memory (LSTM)

Vanishing errors are a problem that LSTM can help with. More than 1000 discrete time steps can be bridged using LSTM. All the units in the hidden layer are swapped out for memory blocks in LSTM networks. There is at least one memory cell in every memory block. One cell in a fundamental LSTM network [34]. The regulatory gates activate the memory cells. These gates regulate the flow of information coming in and leaving. A forget-gate is located between an input gate and an output gate. If the stored data is no longer required, forget gates can reset the state of the linear unit. Simple sigmoid threshold units make up these gates. The range of these activation functions is 0 to 1 [34]. In deep neural models, the dropout [18] process is used to lose neurons randomly throughout each training session. Deep neural networks must go through this procedure to avoid overfitting, a condition in which the network learns to effectively identify variables in fresh samples [18]. In this study, a layer with a 0.2 dropout rate is included. The output $y^{cj}(t)$ of an LSTM memory cell is computed as [34] in Eq. 5:

$$y^{cj}(t) = \mathrm{y^{out_j}(t)}\, h(\mathrm{s_{cj}}(t)), \tag{5}$$

where c stands for internal cell state, j stands for output gate, y^{out_j} is the output of the activation gate, s_{cj} is the internal state of the output gate, and h is the output of the hidden layer.

3.2.4 Gated Recurrent Unit (GRU)

The vanishing problem has been tackled in several ways, but Long-Short-Term Memory (LSTM) is the one that appears in literature the most frequently (Hochreiter and Schmidhuber, 1997) [16]. The use of a set of gates—mechanisms that control learning and memory rates—ensures that long-term data continues to have an impact on recent predictions. However, compared to LSTM, GRU has fewer tensor operations, which speeds up the training process [20].

The update and reset gates are operated by GRUs. It is up to the first one to specify what details about a new entry will be forgotten and what new details will be added. The second, however, talks about how much information from the past or the long term would be forgotten. As was already mentioned, there are various neural networks that determine which information should be retained or forgotten. Because it normalizes its output into values between 0 and 1, the sigmoidal activation function used by both gates to operate helps to streamline the process. Any value multiplied by 0 will therefore be lost; however, values multiplied by 1 will be retained [20].

The cell first multiplies h_{t-1} and x_t together to determine the value of h_t. It then sends the resulting vector to the reset and update gates to determine their respective outputs, r_t and u_t, using Eqs. (6) and (7).

$$r_t = \sigma(w_r \cdot [h_{t-1}, x_t] + b_r), \tag{6}$$

$$u_t = \sigma(w_u \cdot [h_{t-1}, x_t] + b_u), \tag{7}$$

where b_r and b_u are the bias vectors for the neural networks, x_t refers to an input at a time point, h_{t-1} refers to an output at a previous time point, and W_r and W_u are their weight matrices. The result is concatenated with x_t and sent into a third neural network with a hyperbolic tangent (tanh) activation function after performing a pointwise multiply between r_t and h_{t-1}. Tanh normalizes data between 1 and − 1, controlling the neural network's output and avoiding data that is too large or too small between iterations. This neural network's output $\hat{\boldsymbol{h}}_t$ is calculated using Eq. (8).

$$\widehat{h_t} = tanh\,(w_0[r_t h_t - 1, x_t] + b_0), \tag{8}$$

3.3 Feature Selection Techniques

The process of selecting discriminative feature variables that are most beneficial to the target variable is known as feature selection [35–44]. By eliminating the factors that don't affect how the target variable is determined, the feature selection procedure aims to lower the computing cost of the model.

To select features, a selection wrapper method such as sequential forward (SFS) was applied. Wrapper approaches evaluate the variable subset using the predictor's performance as the objective function and the predictor as a black box, as shown in Fig. 3. The wrapper approaches are generally categorized as sequential selection algorithms. The sequential selection methods begin with an empty set (a complete set), add features, and then subtract features until the maximum objective function is reached. To expedite the selection process, a criterion is selected that gradually raises the objective function until the maximum is obtained with the fewest features. Sequential selection algorithms were used. Given that these algorithms are iterative, they are regarded as sequential [20]. In the proposed approaches, RF-based sequential forward selection (RF-SFS) was implemented. The tuning parameter that was used for RF to achieve high accuracy is max_depth = 20 because the values of the max_depth parameter that go between 5 and 10 achieve an accuracy between 74.1% and 77.9% since the tree is underfit. This led us to increase the value of the max_depth parameter to gain more accurate results. The accuracy remained the same even after passing the max_depth of 20.

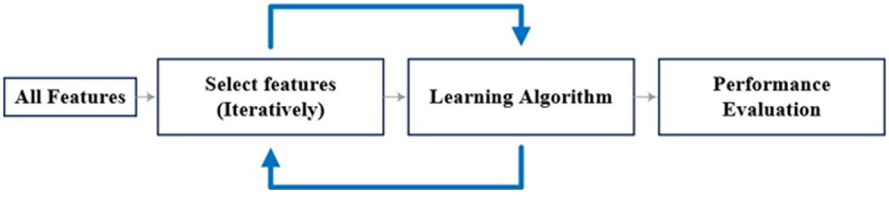

Fig. 3 Wrapper method diagram [35]

4 Proposed SAT-IDSs

Inspired by ML/DL techniques, four hybrid IDS techniques are proposed. The first technique depends on using RF combined by SFS (RF-SFS), the second technique depends on using ANN combined by SFS (RF-SFS-ANN), the third technique depends on using LSTM combined by SFS (RF-SFS-LSTM), and the fourth technique depends on using GRU combined by SFS (RF-SFS-GRU). A subset of high-performing features was chosen from the datasets STIN for satellite networks and UNSW-NB15 for terrestrial networks by using sequential forward selection (RF-SFS) to reduce the complexity and dimensionality of the features. As a result, 10 features were selected from the whole set of features for each dataset. Therefore, the selected features are combined with ML and DL models to detect intrusions.

4.1 Data Pre-processing

Figure 4 shows the dataset preprocessing steps. For the learning process to be successful, this phase is essential. The five processes of data pre-processing are cleaning, minority removal, label encoder, normalization, and feature selection. The Sequential Forward Selection (SFS) feature selection method was used to reduce the dimensionality of the dataset. This technique is implemented based on Random Forest (RF) to select the best subset.

This section discusses the following data preprocessing steps:

1. Cleaning: There are 45 and 32 features in the UNSW-NB15 and STIN datasets, respectively. In the UNSW-NB15 dataset, two features are the attack's class designations, and 43 of them are important features. While "label" is a binary class label, "attack cat" is a multi-class label. The term "label" was eliminated because the proposed ML and DL models are built to conduct multi-classification for intrusion detection. In the STIN dataset, one feature is class designations, and 31 of them are important features. While "label" is a multi-class label.
2. Minority Removal: The performance of machine learning and deep learning might be negatively impacted by extremely unbalanced datasets. In the UNSW-

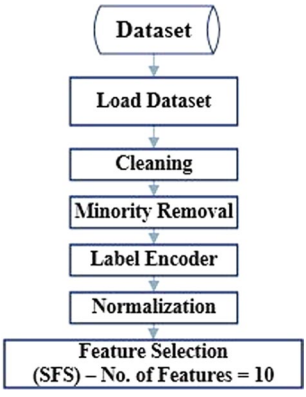

Fig. 4 Preprocessing dataset steps

NB15 dataset, four minority classes are eliminated since they made up 1.141%, 0.996%, 0.646%, and 0.074% of the training set, respectively. These classes were "Analysis," "Backdoor," "Shellcode," and "Worms." But in the STIN dataset, four minority classes are merged into one DDoS class, and another three minority classes are merged into one Botnet class because the minority classes are a subset of the main class, and the main target in this dataset set is satellite attacks. Finally, a balanced training dataset for UNSW-NB15 was produced that includes DoS, Exploits, Fuzzers, Generic, Normals, and Reconnaissance with 4,089, 11,132, 6,062, 18,871, 37,000, and 3,496 records, respectively. Also, a balanced training dataset for STIN was produced that included Botnet, DDoS, Syn_DDoS, and UDP_DDoS with 40,401, 57,292, 54,789, and 57,082 records, respectively.

3. Label Encoder: The UNSW-NB15 dataset includes three categorical characteristics that each contain categorical values: "service", "proto", "state" and "attack_cat". Using a label encoder, these features were converted from string values to integers [31], but the STIN dataset has only one categorical characteristic in the "Label" feature.
4. Normalization: The value range of each feature can be unified through normalization. To convert the range of feature values between 0 and 1, "MinMax" Normalization was utilized. According to Eq. 9 [32], the difference between the minimum value and the scale size is used to determine the new value.

$$\overline{x_i} = \frac{x_i - min(x_i)}{(x_i) - min(x_i)}, \tag{9}$$

where $\overline{x_i}$ is normalized value, $\boldsymbol{x_i}$ represents the ***ith*** feature vector, $\boldsymbol{min}(\boldsymbol{x_i})$ returns the minimum value of the vector, and $(\boldsymbol{x_i})$ returns the maximum value of the vector.

5. RF-SFS: sequential forward feature selection (SFS) based on a random forest model (RF-SFS) was applied to select the best subset with ten features to achieve high accuracy. These features are "proto, dbytes, sttl, dttl, dload, swin, synack, smean, dmean, ct_dst_src_ltm" for the UNSW-NB15 dataset and "fl_dur, l_fw_pkt, pkt_len_min, pkt_len_max, fl_byt_s, fw_hdr_len, fin_cnt, urg_cnt, fw_pkt_blk_avg, fw_win_byt" for the STIN dataset.

4.2 Training and Testing Set Preparation

The training set is used to fit the model, and the test set is used to verify the model's performance. The preprocessing steps of the training and testing sets are detailed in Fig. 5. The UNSW-NB15 dataset has 82,332 records for the training set and 175,341 records for the testing set, but the STIN dataset has 209,564 records for the training set and 41,913 records for the testing set. The training and testing set details are shown in Table 2.

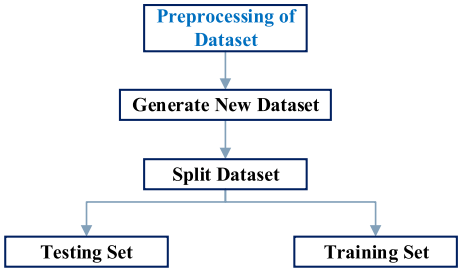

Fig. 5 Training and test preparation step

4.3 Implemented IDS Models

In this section, ML/DL approaches-based RF-SFS was proposed in detail. This step is executed after two stages, mainly the preprocessing of the dataset and the dataset splitting, as shown in Fig. 6. In the block of model training, one of the ML- or DL-based RF-SFS approaches was implemented, and then the model was evaluated.

4.3.1 RF-SFS-RF Model

The RF-SFS-RF approach was implemented using an RF classifier with selected features from the SFS-RF method and achieved high accuracy results with max_depth = 20. The time complexity of RF in the proposed model = O *(depth of tree*k)*. Where k is the number of decision trees and $O(d)$ is the complexity of SFS. Where d is the desired number of features. The overall complexity of SFS-RF is O *(2*(depth of tree*k))* + $O(d)$.

4.3.2 RF-SFS-ANN Model

The RF-SFS-ANN approach was implemented with the "Adam" optimizer, and it included three hidden layers with the same number of neurons (50), an input layer with input dimension parameters of 10 for UNSW-NB15, and STIN datasets that are equal to input features. The used activation function in the input and hidden layers is

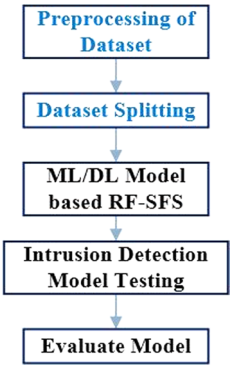

Fig. 6 Proposed models

tanh; the output layer has 6 and 4 neurons for the UNSW-NB15 and STIN datasets, respectively, which are equal to the number of output classes, and the used activation function is "softmax".

For ANN, there are n input features, 3 hidden layers with m_i neurons for each layer, k output neurons with t training samples, and n epochs. So, the time complexity of ANN is $O\ (nt*(n*m_1+m_1*m_2+m_2*m_3+m_3*k))$. Therefore, the overall complexity of RF-SFS-ANN is $O\ (nt\ *\ (n*m_1+m_1*m_2+m_2*m_3+m_3*k))+O\ (depth\ of\ tree*k)+O\ (d)$.

4.3.3 RF-SFS-LSTM Model

The RF-SFS-LSTM approach was implemented with the "Adam" optimizer, and it included three hidden layers with the same number of neurons (50): an input layer with input dimension parameters of 10 for UNSW-NB15 and STIN datasets, which are equal to input features; an output layer with 6 and 4 neurons for UNSW-NB15 and STIN datasets, respectively, which are equal to the number of output classes; and an activation function of "softmax".

LSTM is local in space and time, which means that the input length does not affect the storage requirements of the network, and for each time step, the time complexity per weight is $O\ (1)$. Therefore, the complexity of an LSTM per time step is equal to $O(w)$, where w is the total number of weights in the network. So, the overall complexity of RF-SFS-LSTM $= O\ (depth\ of\ tree*k)+O(d)+O(w)$.

4.3.4 RF-SFS-GRU Model

This approach was designed with three hidden layers with 50 neurons: an input layer with 10 neurons and an output layer with 6 and 4 neurons, respectively, for the UNSW-NB15 and STIN datasets. The activation function was "Softmax" and the "Adam" optimizer was used when fitting the model.

In the proposed RF-SFS-GRU, there are I input features, 3 hidden layers (H, J, and N) with m_i neurons for each layer, and k output neurons with t training samples and n epochs. So, the time complexity of GRU is $O\ (nt*(IH+J^2+N^2+N*k))$ and the overall time complexity of RF-SFS is $O\ (depth\ of\ tree*k)+O(d)$, as mentioned previously. Therefore, the overall complexity of RF-SFS-GRU is $O\ (nt*(IH+J^2+N^2+N*k))+O\ (depth\ of\ tree*k)+O\ (d)$.

5 Experimental Results

Kaggle [44] was used as the experimental environment. Kaggle notebooks run in a remote computational environment. It has 20 gigabytes of auto-saved disc space (/kaggle/working).

- The CPU specifications are: 4 CPU cores and 30 gigabytes of RAM.
- GPU specifications are: 1 Nvidia, 2 CPU cores, and 13 gigabytes of RAM.

Table 3 Simplified confusion matrix [5]

		Predicted class	
		Positive	Negative
Actual class	Positive	TP	FN
	Negative	FP	TN

Scikit-Learn, Numpy, Pandas, Matplotlib, and other packages were also used to provide data processing, feature selection, and visualization functions [44] to analyze the performance of the proposed SAT-IDSs.

5.1 Performance Metrics

To evaluate the performance of the four proposed SAT-IDSs, performance metrics like accuracy, recall, precision, and F1-score were calculated.

Table 3 presents a simplified confusion matrix that differentiates the classification results [5]. Based on the one versus all principle, there are generally four cases in ML/DL classification tasks where:

- True Positive (TP): represents correctly classified positive samples.
- False Negative (FN): represents incorrectly classified positive samples.
- False Positive (FP): represents incorrectly classified negative samples.
- True Negative (TN): represents correctly classified negative samples.

The equation's definition of accuracy determines the proportion of correctly classified samples to all samples, as defined in Eq. 10 [40].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{10}$$

Recall calculates the ratio of correctly classified positive samples to all samples that were supposed to be positive, as defined in Eq. 11 [40].

$$Recall\,(True\ Positive\ Rate) = \frac{TP}{TP + FN}, \tag{11}$$

Precision, as defined in Eq. 12, returns the ratio of classified positive samples to all samples that are predicted to be positive [40].

$$Precision = \frac{TP}{TP + FP}, \tag{12}$$

The *F1-score,* as defined in Eq. 13, returns the harmony mean of recall and precision. In multi-class imbalanced data, it can be applied as a performance metric to address recall and precision faults [40].

$$F1\text{–}Score = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right), \tag{13}$$

5.2 Performance Analysis

To prove the efficiency of the proposed IDSs, the proposed approaches were tested on two datasets: UNSW-NB15 and STIN. Classifier performance evaluation is commonly computed using evaluation metrics such as accuracy, precision, recall, and the F1 score. These metrics are measured using a confusion matrix, as given in Eqs. 10, 11, 12, and 13.

To test the performance of individual attack labels on the UNSW-NB15 dataset, a confusion matrix for the RF-SFS-LSTM multiclass classification model was computed, as shown in Fig. 7. The RF-SFS-LSTM model performs well for Reconnaissance, Generic, Exploits, and Normal classes, as demonstrated by the darker squares in the confusion matrix. The performance of a model for classification at all classification thresholds was evaluated using a ROC curve to evaluate the model's robustness. The ROC curve for the RF-SFS-LSTM multiclass classification model is shown in Fig. 8. While the model is unable to distinguish between positive and negative class values for the Fuzzers class, which has the worst ROC curve, the proposed approach identifies all the classes with better ROC curves.

Figure 9 displays the confusion matrix for the RF-SFS-GRU multiclass classification model on the UNSW-NB15 dataset, which was used to evaluate the performance of individual attack labels. Through darker squares, it performs well for Reconnaissance, Generic, Exploits, and Normal classes. The ROC curve for the

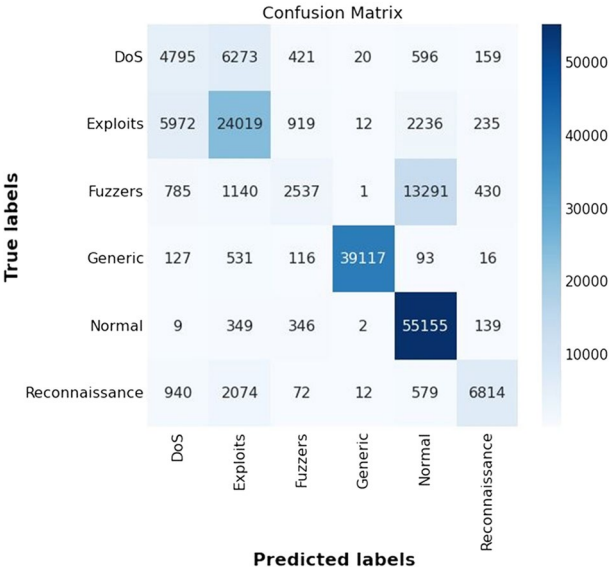

Fig. 7 Confusion matrix of test set for RF-SFS-LSTM (UNSW-NB15)

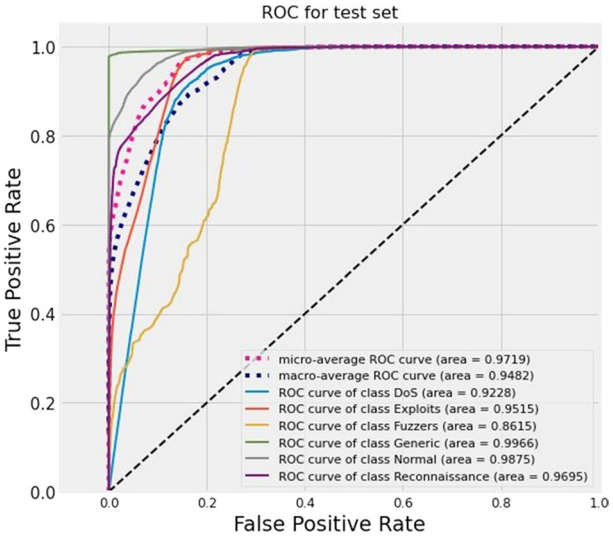

Fig. 8 ROC curve for RF-SFS-LSTM (UNSW-NB15)

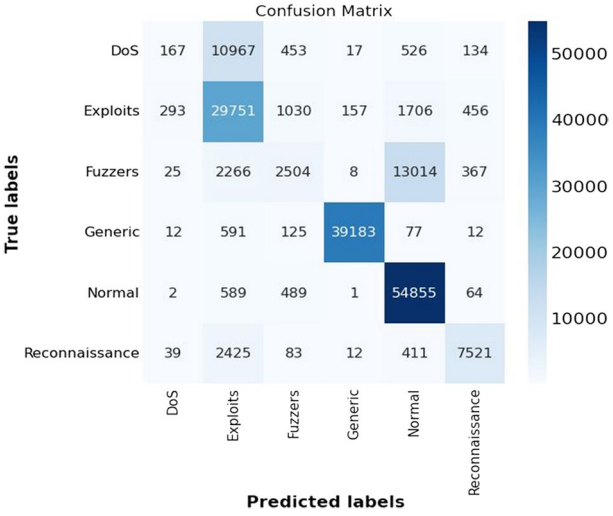

Fig. 9 Confusion matrix of test set for RF-SFS-GRU (UNSW-NB15)

RF-SFS-GRU multiclass classification model is shown in Fig. 10. This proposed model outperforms the RF-SFS-LSTM model in detecting Fuzzers.

Figure 11 illustrates the confusion matrix for the RF-SFS-LSTM multiclass classification model on the STIN dataset. Based on the darker squares in the confusion matrix of the RF-SFS-LSTM classifier, it performs well for terrestrial attacks, which include Botnet and DDos categories, but it performs better for the Syn_DDos class than the UDP_DDos class in satellite attacks, which include Syn_DDos and

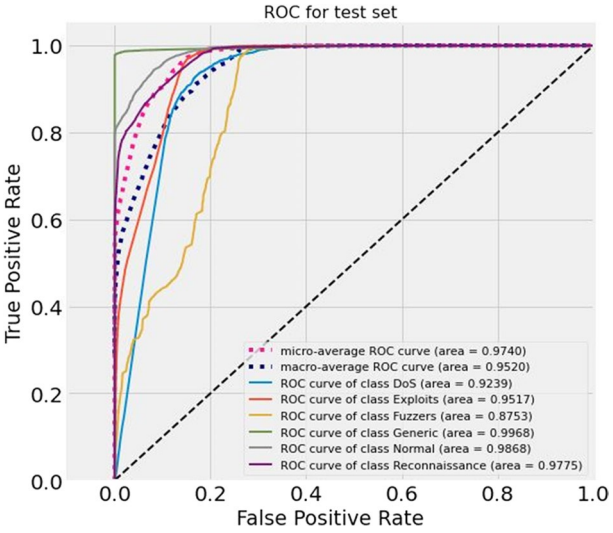

Fig. 10 ROC curve for RF-SFS-GRU (UNSW-NB15)

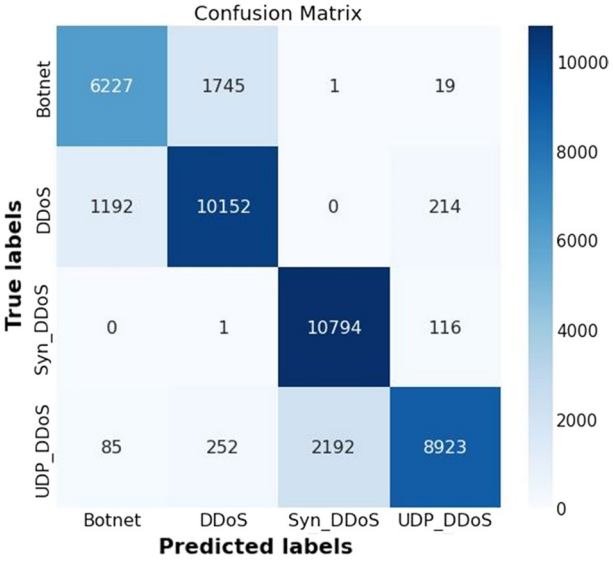

Fig. 11 Confusion Matrix of test set for RF-SFS-LSTM (STIN)

UDP_DDos attacks. The ROC curves for the RF-SFS-LSTM multiclass classification model in the STIN dataset are shown in Fig. 12. The model distinguishes the Syn_DDos class better than the UDP_DDos class.

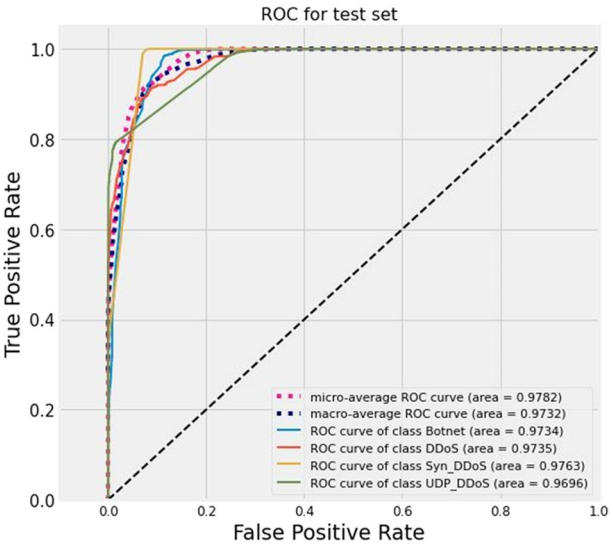

Fig. 12 ROC curve for RF-SFS-LSTM (STIN)

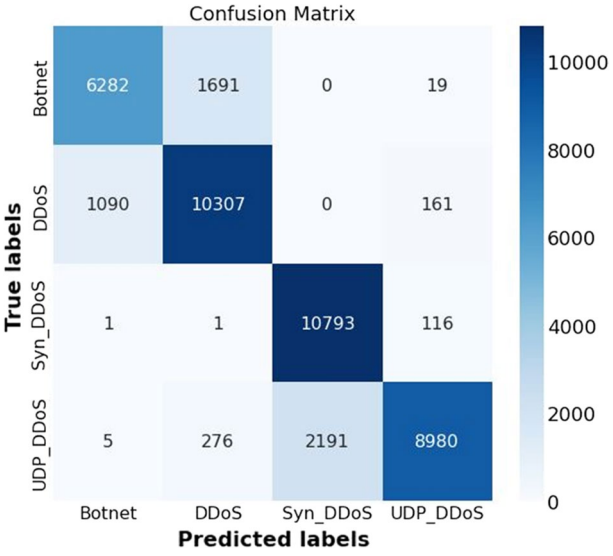

Fig. 13 Confusion matrix of test set for RF-SFS-GRU (STIN)

The confusion matrix for the RF-SFS-GRU multiclass classification model on the STIN dataset is shown in Fig. 13. It is obvious that it outperforms the RF-SFS-LSTM multiclass classification model for the UDP_DDos class. The ROC curve for the RF-SFS-GRU multiclass classification model in the STIN dataset

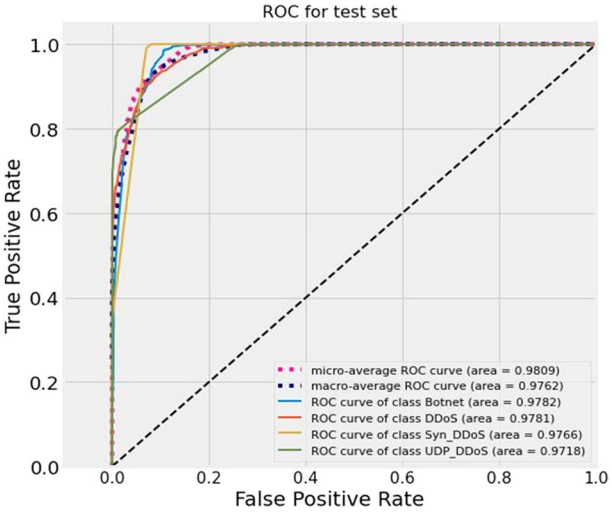

Fig. 14 ROC curve for RF-SFS-GRU (STIN)

is shown in Fig. 14. This model outperforms the RF-SFS-LSTM model in distinguishing the Syn_DDos class.

5.3 Performance Comparison

To prove the efficacy of the proposed IDSs, they were tested on two datasets, UNSW-NB15 and STIN, and compared to others. Table 4 and Fig. 15 demonstrate the experimental results for deep learning models like ANN, GRU, and LSTM and machine learning models like RF before utilizing the RF-SFS approach on the UNSW-NB15 dataset. The experimental results for proposed approaches based on deep learning, such as RF-SFS-ANN, RF-SFS-GRU, and RF-SFS-LSTM, and machine learning models, such as RF-SFS, are shown in Table 5 and Fig. 16, and they achieved higher accuracy results using SFS technique-based RF classifiers than those in [18], which used 19 selected features. The highest accuracy and precision

Table 4 Results of classifiers on UNSW-NB15 dataset—multiclass classification

ML/DL model	Testing acc. (%)	Precision (%)	Recall (%)	F1-Score (%)
RF	75.4	66	46	47
ANN	69.05	71.61	65.33	68.32
GRU	74	55	44	42
LSTM	75	54	40	40
Proposed Approach In [18] based ANN	75.62	79.92	75.61	76.58
Proposed Approach In [18] based KNN	70.09	75.79	70.21	72.03

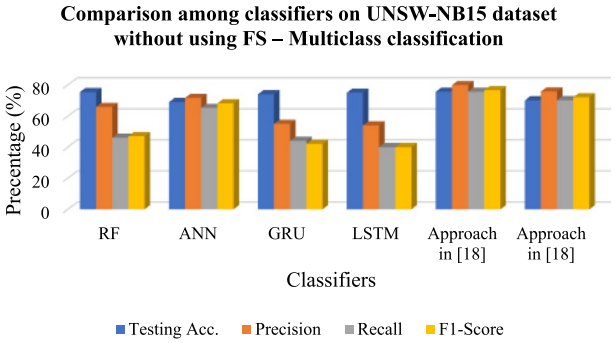

Fig. 15 Comparison among classifiers without using a FS technique (UNSW-NB15 dataset)

Table 5 Results of classifiers with FS technique on UNSW-NB15 dataset—multiclass classification

ML/DL model	Testing acc. (%)	Precision (%)	Recall (%)	F1-Score (%)
SFS-RF	78.52	72	66	68
RF-SFS-ANN	78.23	85.13	68.85	76.13
RF-SFS-GRU	79	67	63	62
RF-SFS-LSTM	78	70	63	64
ANN [18] using 19 features	77.51	79.50	77.53	77.28
KNN [18] using 19 features	72.30	77.24	72.30	73.81

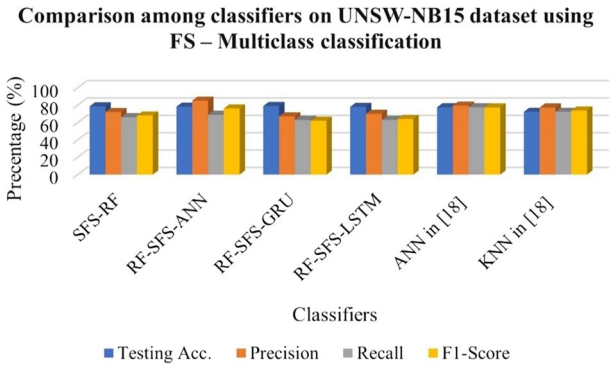

Fig. 16 Comparison among classifiers using FS technique (UNSW-NB15 dataset)

in Table 4 were obtained utilizing the RF-SFS-GRU and RF-SFS-ANN approaches rather than those in [18].

The performance of the proposed IDSs when applied to the whole STIN dataset without feature selection is shown in Table 6 and Fig. 17. The RF technique produces the best results. The average results for proposed techniques utilizing the RF-SFS feature selection method are shown in Table 7 and Fig. 18. RF-SFS significantly

Table 6 Average results of classifiers on STIN dataset—multiclass classification

ML/DL model	Testing acc. (%)	Precision (%)	Recall (%)	F1-Score (%)
RF	85.41	82.54	81.11	81
GRU	79	75	71	71
LSTM	82	77	75	75
ANN	63.61	69.16	61.61	65.17

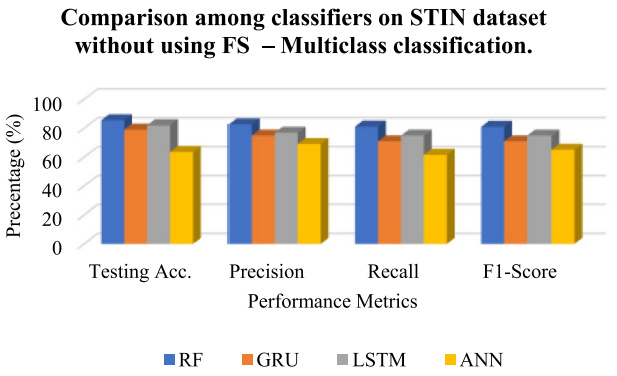

Fig. 17 Comparison among classifiers without using RF-FS technique (STIN dataset)

Table 7 Average results of classifiers using SFS-RF on STIN dataset—multiclass classification

ML/DL model	Testing acc. (%)	Precision (%)	Recall (%)	F1-Score (%)
SFS-RF	90.5	91.19	90.41	90
RF-SFS-GRU	87	87	86	86
RF-SFS-LSTM	86	86	85	85
RF-SFS-ANN	71.47	71.51	71.39	71.44

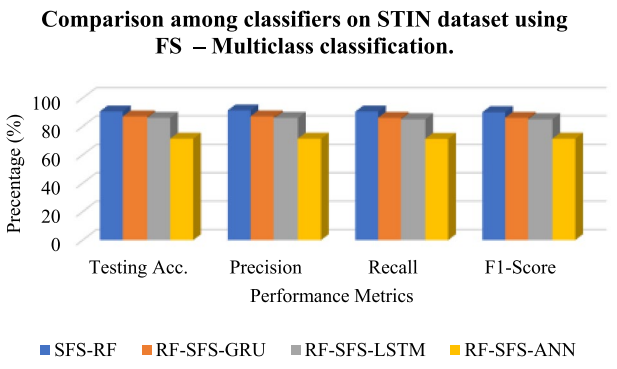

Fig. 18 Comparison among classifiers using RF-FS technique (STIN dataset)

Table 8 Accuracy of classifiers on STIN dataset

Attack type	RFMLP [5]	RF-SFS-LSTM	RF-SFS-GRU
UDP_DDos	100.0	92.81	93.36
Syn_DDos	93.18	94.49	94.49

outperforms other proposed approaches; as shown in Table 6, utilizing the RF-SFS feature selection approach improves the performance of all proposed IDSs.

Table 8 compares the performance of the proposed IDSs with the previous IDSs proposed in [5] based on satellite attacks in the STIN dataset. It concludes that the RF-SFS-LSTM and RF-SFS-GRU approaches outperformed [5] in terms of accuracy in Syn_DDos attacks, which are highlighted in grey in Table 8. As shown in Table 8, IDS in [5] produced a higher accuracy result in a UDP_DDos attack than the proposed approaches.

6 Discussion

The previous section presented a performance analysis and comparison of the four ML/DL-based RF-SFS techniques. The RF-SFS feature selection technique is exploited to significantly reduce the execution time of training and testing data by minimizing the number of selected features while enhancing the accuracy of intrusion detection. It selects 10 features among 32 features and 45 features using the STIN and UNSW-NB15 datasets, respectively.

When applying to the STIN dataset, the accuracy of the RF classifier increased by using RF-SFS from 85.41% to 90.50%, and the accuracy of the ANN classifier increased from 63.61% to 71.47%. Furthermore, by using RF-SFS with an LSTM classifier, the accuracy increased from 82 to 86%. In the last approach, the accuracy of the GRU classifier increased from 79 to 87%. The experimental results demonstrate that the RF-SFS-GRU classifier achieved a higher accuracy score in the UDP_DDoS attack than the RF-SFS-LSTM classifier. In addition, RF-SFS-LSTM and RF-SFS-GRU IDS recorded higher detection accuracy for Syn_DDos than RFMLP [5]. When applied to the UNSW-NB15 dataset, the accuracy of the RF classifier increased by using SFS-RF from 75.4% to 78.5%, and the accuracy of the ANN classifier increased from 69.05 to 78.23%. Furthermore, by using RF-SFS with an LSTM classifier, the accuracy increased from 75 to 78%. In the last approach, the accuracy of the GRU classifier increased from 74 to 79%. The experimental results also demonstrate that the RF-SFS-GRU classifier achieved higher accuracy results than other proposed approaches. The proposed RF-SFS-GRU and RF-SFS-ANN IDS achieved higher accuracy and precision, respectively, than the approaches in [18].

7 Conclusion and Future Works

The requirement for cybersecurity solutions to prevent attacks in the modern network environment has increased along with the number of network intrusion attacks. To provide a high level of security for both satellite and terrestrial networks, four ML/DL based hybrid IDSs are proposed in this paper. The proposed models are evaluated and verified using the UNSW-NB15 and STIN datasets. The RF-SFS feature selection technique is utilized to improve the classification results and reduce the execution time. The proposed models ensure their efficiency when compared to the literature's existing models [5–18]. When applied to the UNSW-NB15 dataset, SFS-RF and RF-SFS-GRU approaches achieved high accuracy scores of 78.52% and 79%, respectively, rather than the proposed approach in [18], which achieves accuracy scores in the ANN and KNN approaches of 72.30% and 77.51%, respectively, when using 19 features. In the STIN dataset, SFS-RF and RF-SFS-GRU achieved high accuracy results of 90.5% and 87%, respectively. The proposed approaches achieved high accuracy in Syn_DDos attacks compared to [5], but the RFMLP model [5] achieved high accuracy in UDP_DDos attacks compared to the proposed approaches. In the future, a new dataset with more various types of satellite attacks and normal traffic will be constructed to test the efficiency of the proposed models.

Acknowledgements The authors would like to thank Norwegian University of Science and Technology, Norway for funding the Article Processing Charges (APCs) of this publication. The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia for their support. Special acknowledgments are given to Automated Systems & Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia.

Author Contributions Conceptualization, ATA, AMM and SAE; Data curation, ES and IAH; Formal analysis, ATA, ES, AMM, IAH and SAE; Funding acquisition, IAH; Investigation, ATA and SAE; Methodology, ATA, ES, AMM, IAH and SAE; Resources, ES, AMM, IAH and SAE; Software, ES; Supervision, ATA, AMM and SAE; Validation, ATA, AMM, IAH and SAE; Visualization, ATA; Writing – review & editing, ATA, ES, AMM, IAH and SAE.

Funding Open access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital). This research was funded by Norwegian University of Science and Technology, Norway.

Data Availability Not applicable.

Declarations

Conflict of interest The authors declare no conflict of interest.

Ethical Approval Not applicable.

Informed Consent Not applicable.

References

1. Rath, M., Mishra, S.: Security approaches in machine learning for satellite communication. In: Machine Learning and Data Mining in Aerospace Technology, pp. 189–204 (2020)
2. Li, K., Zhou, H., Tu, Z., Wang, W., Zhang, H.: Distributed network intrusion detection system in satellite-terrestrial integrated networks using federated learning. IEEE Access **8**, 214852–214865 (2020)
3. Nguyen, N.T., Chang, C.C.: A biometric-based authenticated key agreement protocol for user-to-user communications in mobile satellite networks. Wirel. Pers. Commun. **107**(4), 1727–1758 (2019)
4. Magdy, M.E., et al.: Anomaly-based intrusion detection system based on Feature selection and Majority Voting. Indones. J. Electr. Eng. Comput. Sci. (2023). https://doi.org/10.11591/ijeecs.v30.i3.pp1699-1706
5. Ashraf, I., Narra, M., Umer, M., Majeed, R., Sadiq, S., Javaid, F., Rasool, N.: A deep learning-based smart framework for cyber-physical and satellite system security threats detection. Electronics **11**(4), 667 (2022)
6. Elsaid, S.A., Albatati, N.S.: An optimized collaborative intrusion detection system for wireless sensor networks. Soft Comput. **24**(16), 12553–12567 (2020)
7. Elsayed, R., Hamada, R., Hammoudeh, M., Abdalla, M., Elsaid, S.A.: A hierarchical deep learning-based intrusion detection architecture for clustered Internet of Things. J. Sens. Actuator Netw. **12**(1), 3 (2022)
8. Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. IEEE Access **7**, 41525–41550 (2019)
9. Ahmed, L.A.H., Hamad, Y.A.M.: Machine learning techniques for network-based intrusion detection system: a survey paper. In: National Computing Colleges Conference (NCCC). IEEE, 2021.
10. Eshakagdy, M., Matter, A.H.M.E.D., Hussin, S., Hassan, D., Elsaid, S.: A Comparative study of intrusion detection systems applied to NSL-KDD Dataset. Egypt. Int. J. Eng. Sci. Technol. (2022). https://doi.org/10.21608/eijest.2022.137441.1156
11. Stiawan, D., Idris, M.Y.B., Bamhdi, A.M., Budiarto, R.: CICIDS-2017 dataset feature analysis with information gain for anomaly detection. IEEE Access **8**, 132911–132921 (2020)
12. Maseer, Z.K., Yusof, R., Bahaman, N., Mostafa, S.A., Foozy, C.F.M.: Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. IEEE Access **9**, 22351–22370 (2021)
13. Shrestha, R., Omidkar, A., Roudi, S.A., Abbas, R., Kim, S.: Machine-learning-enabled intrusion detection system for cellular connected UAV networks. Electronics **10**(13), 1549 (2021)
14. Jiang, K., Wang, W., Wang, A., Wu, H.: Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access **8**, 32464–32476 (2020)
15. Ahmad, I., et al.: An efficient network intrusion detection and classification system. Mathematics **10**(3), 530 (2022)
16. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Comput. Electr. Eng. **40**(1), 16–28 (2014)
17. Kasongo, S.M., Sun, Y.: Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. J. Big Data **7**, 1–20 (2020)
18. Assis, M.V., Carvalho, L.F., Lloret, J., Proença, M.L., Jr.: A GRU deep learning system against attacks in software defined networks. J. Netw. Comput. Appl. **177**, 102942 (2021)
19. Thakkar, A., Lohiya, R.: Attack classification using feature selection techniques: a comparative study. J. Ambient. Intell. Humaniz. Comput. **12**, 1249–1266 (2021)
20. Gaur, V., Kumar, R.: Analysis of machine learning classifiers for early detection of DDoS attacks on IoT devices. Arab. J. Sci. Eng. **47**(2), 1353–1374 (2022)

21. Moustafa, N., Slay, J.: UNSW-NB15. A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), November 2015, pp. 1–6. IEEE.
22. Li, M., Zhang, W., Hu, B., Kang, J., Wang, Y., Lu, S.: Automatic assessment of depression and anxiety through encoding pupil-wave from HCI in VR scenes. ACM Trans. Multimedia Comput. Commun. Appl. (2022). https://doi.org/10.1145/3513263
23. Song, Y., Zhao, G., Zhang, B., Chen, H., Deng, W., Deng, W.: An enhanced distributed differential evolution algorithm for portfolio optimization problems. Eng. Appl. Artif. Intell. **121**, 106004 (2023)
24. Zhou, X., Cai, X., Zhang, H., Zhang, Z., Jin, T., Chen, H., Deng, W.: Multi-strategy competitive-cooperative co-evolutionary algorithm and its application. Inf. Sci. **635**, 328–344 (2023)
25. Chaganti, R., Suliman, W., Ravi, V., Dua, A.: Deep learning approach for SDN-enabled intrusion detection system in IoT networks. Information **14**(1), 41 (2023)
26. Henry, A., Gautam, S., Khanna, S., Rabie, K., Shongwe, T., Bhattacharya, P., Sharma, B., Chowdhury, S.: Composition of hybrid deep learning model and feature optimization for intrusion detection system. Sensors **23**(2), 890 (2023)
27. Mijalkovic, J., Spognardi, A.: Reducing the false negative rate in deep learning based network intrusion detection systems. Algorithms **15**(8), 258 (2022)
28. Sahu, S.K., Mohapatra, D.P., Rout, J.K., Sahoo, K.S., Pham, Q.V., Dao, N.N.: A LSTM-FCNN based multi-class intrusion detection using scalable framework. Comput. Electr. Eng. **99**, 107720 (2022)
29. An, P., Wang, Z., Zhang, C.: Ensemble unsupervised autoencoders and Gaussian mixture model for cyberattack detection. Inf. Process. Manag. **59**(2), 102844 (2022)
30. Tonkal, Ö., Polat, H., Başaran, E., Cömert, Z., Kocaoğlu, R.: Machine learning approach equipped with neighbourhood component analysis for DDoS attack detection in software-defined networking. Electronics **10**(11), 1227 (2021)
31. Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., Kwak, J.: IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 Dataset. J. Big Data **10**(1), 1–26 (2023)
32. Subba, B., Biswas, S., Karmakar, S.: A neural network based system for intrusion detection and attack classification. In: 2016 Twenty Second National Conference on Communication (NCC), IEEE, March 2016, pp. 1–6
33. Muhuri, P.S., Chatterjee, P., Yuan, X., Roy, K., Esterline, A.: Using a long short-term memory recurrent neural network (LSTM-RNN) to classify network attacks. Information **11**(5), 243 (2020)
34. Halbouni, A., Gunawan, T.S., Habaebi, M.H., Halbouni, M., Kartiwi, M., Ahmad, R.: CNN-LSTM: hybrid deep neural network for network intrusion detection system. IEEE Access **10**, 99837–99849 (2022)
35. Faker, O., Dogdu, E.: Intrusion detection using big data and deep learning techniques. In: Proceedings of the 2019 ACM Southeast Conference, April 2019, pp. 86–93.
36. Aziz, A.S.A., Hassanien, A.E., Azar, A.T., Hanafy, S.E.: Genetic algorithm with different feature selection techniques for anomaly detectors generation. In: 2013 Federated Conference on Computer Science and Information Systems (FedCSIS), Kraków, Poland, 8–11 Sept 2013 (2013).
37. Jothi, G., Inbarani, H.H., Azar, A.T., Devi, K.R.: Rough set theory with Jaya optimization for acute lymphoblastic leukemia classification. Neural Comput. Appl. **31**(9), 5175–5194 (2019)
38. Jothi, G., Inbarani, H.H., Azar, A.T.: Hybrid tolerance rough set: PSO based supervised feature selection for digital mammogram images. Int. J. Fuzzy Syst. Appl. **3**(4), 15–30 (2013)
39. Inbarani, H.H., Banu, P.K.N., Azar, A.T.: Feature selection using swarm-based relative reduct technique for fetal heart rate. Neural Comput. Appl. **25**(3–4), 793–806 (2014). https://doi.org/10.1007/s00521-014-1552-x
40. Khan, A.R., Saba, T., Khan, M.Z., Fati, S.M., Khan, M.U.G.: Classification of human's activities from gesture recognition in live videos using deep learning. Concurr. Comput. Pract. Exp. (2022). https://doi.org/10.1002/cpe.6825
41. Rehman, A., Saba, T., Tariq, U., Noor, A.: Deep learning-based COVID-19 detection using CT and X-ray images: current analytics and comparisons. IT Prof. **23**, 63–68 (2021). https://doi.org/10.1109/MITP.2020.3036820
42. Rehman, A., Sadad, T., Saba, T., Hussain, A., Tariq, U.: Real-time diagnosis system of COVID-19 using X-ray images and deep learning. IT Prof. **23**, 57–62 (2021). https://doi.org/10.1109/MITP.2020.3042379

43. Saba, T., Khan, M.A., Rehman, A., Marie-Sainte, S.L.: Region extraction and classification of skin cancer: a heterogeneous framework of deep CNN Features fusion and reduction. J. Med. Syst. **43**, 289:1-289:19 (2019). https://doi.org/10.1007/s10916-019-1413-3
44. Bisong, E.: Building Machine Learning and Deep Learning Models on Google Cloud Platform, pp. 59–64. Apress, Berkeley, CA (2019)

Ahmad Taher Azar is a full Professor at Prince Sultan University, Riyadh, Saudi Arabia. He is also a Full Professor at the Faculty of Computers and Artificial Intelligence, Benha University, Egypt. He is a leader of Automated Systems & Soft Computing Lab (ASSCL), Prince Sultan University, Saudi Arabia. He is currently an editor for IEEE Systems Journal, IEEE Transactions on Neural Networks and Learning Systems, Springer's Human-centric Computing and Information Sciences, and Elsevier's Engineering Applications of Artificial Intelligence. Prof. Azar has expertise in Artificial Intelligence, Control Theory and Applications, Robotics, Machine Learning, Computational Intelligence, and dynamic system modeling. He has authored/co-authored over 450 research papers in prestigious peer-reviewed journals, book chapters, and conference proceedings.

Esraa Shehab received her B.Sc. in Computer Engineering and Automated Control Systems from the Faculty of Engineering at Tanta University, Tanta, Egypt. M.Sc. degree in design of a hybrid management system for query execution using central and graphical units from Computer Engineering and Automated Control Systems, Faculty of Engineering, Tanta University, Tanta, Egypt She worked as an academic staff member at BUC in Egypt. Currently, she has been a Ph.D. student at Zagazig University since 2021. Her research interests are in communication engineering and artificial intelligence.

Ahmed M. Mattar is a Professor at the Department of Computer Engineering and Artificial Intelligence, Military Technical College, Cairo, Egypt. He received his Ph.D. in Computer Engineering and Intelligent Systems in 2017 from the Electronic and Computer Engineering Department, University of Alberta, Alberta, Canada. His current research interests include; cybersecurity, intrusion detection, ethical hacking, digital forensics, data mining, deep learning, machine learning, bot detection, fake news detection, sentiment analysis, steganography, distributed systems, digital transformation, system engineering, and networking.

Ibrahim A. Hameed is a Professor at the Norwegian University of Science and Technology (NTNU). He is an IEEE senior member and the elected chair of the IEEE Computational Intelligence Society in Norway. He holds a Ph.D. in Artificial Intelligence from Korea University, South Korea, and another Ph.D. in field robotics from Aarhus University, Denmark. He has authored more than 200 journal and conference articles. His current research interests include AI, autonomous systems, and robotics for sustainability

Shaimaa Ahmed Elsaid is an Associate Prof. at the Electronics and Communications Dep., Faculty of Engineering, Zagazig University, Egypt. She received an MSc degree (2006) in Networks Security and PhD degree (2011) in Multimedia Security from the Faculty of Engineering, Zagazig University (Egypt). Her current research interests include cybersecurity, the Internet of Things (IoT), Artificial Intelligence, and Digital Image Processing. She is the author of 2 books and many research papers published in international journals, and conference proceedings. Also, she has supervised many graduation projects, MSc, and PhD theses.

Authors and Affiliations

Ahmad Taher Azar[1,2,3] · Esraa Shehab[4,5] · Ahmed M. Mattar[6] · Ibrahim A. Hameed[7] · Shaimaa Ahmed Elsaid[5]

✉ Ahmad Taher Azar
aazar@psu.edu.sa; ahmad.azar@fci.bu.edu.eg; ahmad_t_azar@ieee.org

✉ Ibrahim A. Hameed
ibib@ntnu.no

Esraa Shehab
eng.esraashehab@gmail.com; Israa.Abdelfatah@buc.edu.eg

Ahmed M. Mattar
a.mattar@ieee.org

Shaimaa Ahmed Elsaid
saelsaid29@gmail.com

1 College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia

2 Automated Systems & Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia

3 Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt

4 Department of Electrical Engineering, Computer Engineering and Systems Program, Badr University in Cairo (BUC), Cairo, Egypt

5 Electronics and Communications Department, Faculty of Engineering, Zagazig University, Zagazig, Egypt

6 Department of Computer Engineering and Artificial Intelligence, Military Technical College, Cairo, Egypt

7 Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Larsgardsvegen, 2, 6009 Alesund, Norway