PLOS ONE

RESEARCH ARTICLE

Advancements in intrusion detection: A lightweight hybrid RNN-RF model

Nasrullah Khan[1], Muhammad Ismail Mohmand[1], Sadaqat ur Rehman[2]*, Zia Ullah[1], Zahid Khan[3], Wadii Boulila[3,4]

1 Department of Computer Science Brains Institute, Peshawar, Pakistan, **2** School of Sciences, Engineering and Environment, University of Salford, Salford, United Kingdom, **3** Robotics and Internet-of-Things Laboratory, Prince Sultan University, Riyadh, Saudi Arabia, **4** RIADI Laboratory, National School of Computer Sciences, University of Manouba, Manouba, Tunisia

* s.rehman15@salford.ac.uk

Citation: Khan N, Mohmand MI, Rehman Su, Ullah Z, Khan Z, Boulila W (2024) Advancements in intrusion detection: A lightweight hybrid RNN-RF model. PLoS ONE 19(6): e0299666. https://doi.org/10.1371/journal.pone.0299666

Editor: Rao Faizan Ali, Universiti Teknologi PETRONAS / University of Management and Technology, MALAYSIA

Received: November 25, 2023

Accepted: February 14, 2024

Published: June 21, 2024

Data Availability Statement: The minimal data set associated with this manuscript is publicly available via Kaggle. The CIC-2017 dataset used in this research can be retrieved via the following URL: https://www.kaggle.com/datasets/sadaqatrehman/cic-data-set The NSL-KDD dataset used in this research can be retrieved via the following URL: https://www.kaggle.com/datasets/sadaqatrehman/intrusion-detection-dataset.

Funding: The author(s) received no specific funding for this work.

Abstract

Computer networks face vulnerability to numerous attacks, which pose significant threats to our data security and the freedom of communication. This paper introduces a novel intrusion detection technique that diverges from traditional methods by leveraging Recurrent Neural Networks (RNNs) for both data preprocessing and feature extraction. The proposed process is based on the following steps: (1) training the data using RNNs, (2) extracting features from their hidden layers, and (3) applying various classification algorithms. This methodology offers significant advantages and greatly differs from existing intrusion detection practices. The effectiveness of our method is demonstrated through trials on the Network Security Laboratory (NSL) and Canadian Institute for Cybersecurity (CIC) 2017 datasets, where the application of RNNs for intrusion detection shows substantial practical implications. Specifically, we achieved accuracy scores of 99.6% with Decision Tree, Random Forest, and CatBoost classifiers on the NSL dataset, and 99.8% and 99.9%, respectively, on the CIC 2017 dataset. By reversing the conventional sequence of training data with RNNs and then extracting features before applying classification algorithms, our approach provides a major shift in intrusion detection methodologies. This modification in the pipeline underscores the benefits of utilizing RNNs for feature extraction and data preprocessing, meeting the critical need to safeguard data security and communication freedom against ever-evolving network threats.

1 Introduction

The remarkable and rapid growth of the Internet-of-Things (IoT) has increasingly drawn the attention of cybercriminals, making it more vulnerable to attacks than ever before [1]. To address these vulnerabilities, a system has been developed [2] that is independent of communication protocols, aimed at simplifying the deployment process and reducing the complexities involved [3]. In our experimental performance analysis, the proposed system demonstrated reliable and consistent effectiveness in detecting and responding to both simulated threats and

Competing interests: The authors have declared that no competing interests exist.

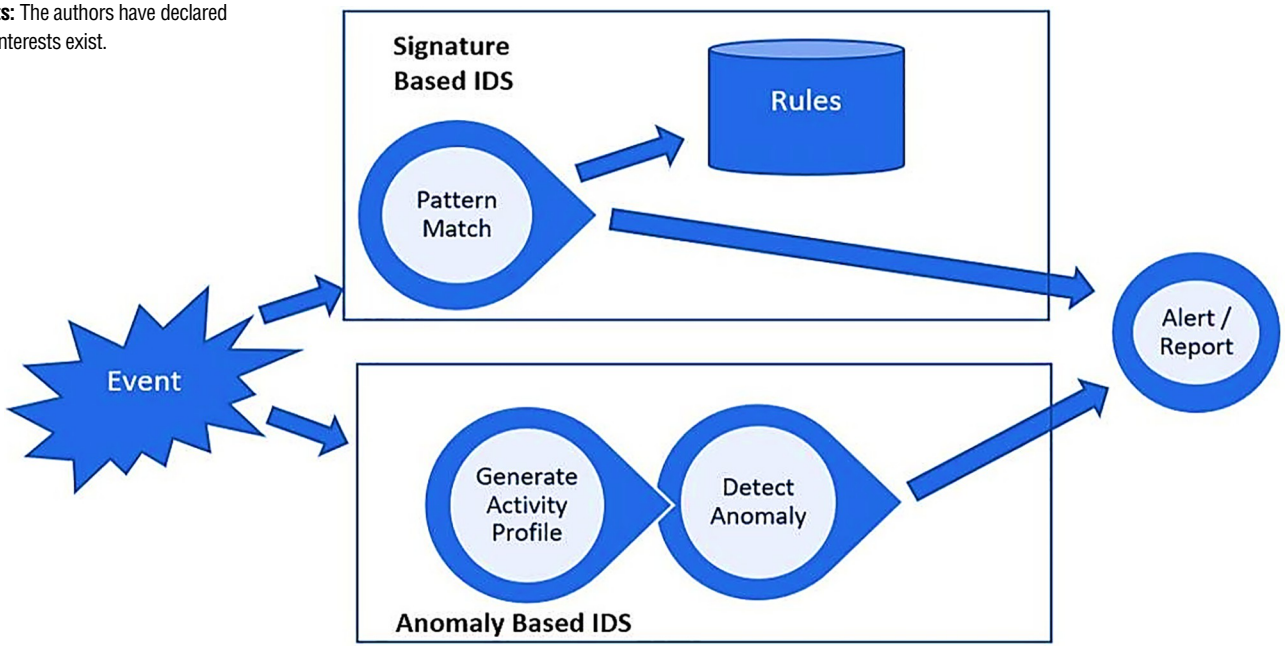

Fig 1. Function of signature-based and anomaly-based IDS.

https://doi.org/10.1371/journal.pone.0299666.g001

real-world invasions. With an average accuracy of 93.74%, it can identify Blackhole, Distributed Denial of Service, Opportunistic Service, Sinkhole, and Workhole attacks. The average precision, recall, and F1-score of the suggested intrusion detection system are 93.71%, 93.82%, and 93.47%, respectively. The 93.21% average detection rate maintained by the cutting-edge deep learning-based intrusion detection system is enough for enhancing the security of IoT networks [4]. The objective of this research is to develop an intrusion detection model by applying various machine learning methods to specific properties identified during the modeling phase [5]. This involves using the NSL-KDD dataset to construct an Intrusion Detection System (IDS). The paper provides a concise review of multiple machine-learning techniques, including Support Vector Machines (SVM), Random Forests, and Decision Trees. These techniques aim to increase detection accuracy by analyzing attacks on the selected dataset, focusing on carefully selected key features. Additionally, the study includes a comparative analysis of these algorithms to determine the most effective approach for intrusion detection. Fig 1 shows an overview of signature-based and anomaly-based intrusion detection systems.

In recent years, the literature has increasingly emphasized the application of Machine Learning (ML) [6] techniques to enhance the accuracy of Intrusion Detection Systems (IDS) [7]. ML is a branch of science that seeks to replicate human learning [8]. Specifically, within the realm of Intrusion Detection Systems (IDS), ML is capable of understanding the patterns of both normal and malicious network traffic. This understanding allows it to effectively differentiate between benign and harmful activities [9]. Recent research indicates that IDS which employ ML algorithms are capable of achieving superior accuracy rates, surpassing those of conventional approaches [10, 11]. However, a common drawback of IDS is its inability to detect previously unseen attacks, particularly those classified as zero-day attacks [12, 13]. A zero-day attack happens when hackers take advantage of a system vulnerability that the developer is unaware of or has not yet been fixed [14, 15]. IDS frequently has the limitation of only being able to detect zero-day attacks and unseen attacks, despite having great accuracy in

identifying known attack activity [16, 17]. Machine learning is an appropriate choice to fix the aforementioned flaw. Drawing inspiration from the comprehensive analysis [18], which explores various machine learning techniques such as decision tree (DT), random forest (RF), support vector machine (SVM), naive Bayes (NB), artificial neural network (ANN) [19, 20], and deep neural network (DNN) [21] for simulating zero-day attacks, our study aims to build upon and extend their findings. We introduce novel contributions in the realm of intrusion detection systems as follows:

- We make a significant contribution by using RNNs for feature extraction before any feature selection, which has the potential to identify intrusion patterns that may not be explicitly defined in traditional handcrafted features.
- Selection of numerous datasets for the evaluation, including the NSL-KDD and the CIC-IDS2017.
- Our approach's ability to maintain or improve accuracy while reducing processing time demonstrates practical value for deployment in resource-constrained environments.
- We also employed various feature selection methods and determined the most effective feature selection method.

Given the limitations inherent in traditional signature-based methods and the challenges posed by false positives in anomaly-based detection, our research investigates a pivotal question: Can the accuracy and efficiency of intrusion detection systems, particularly in resource-constrained environments, be significantly enhanced by employing RNNs for feature extraction before feature selection? This approach aims to unveil complex intrusion patterns that elude detection by conventional handcrafted features, potentially revolutionizing how systems identify and respond to security threats.

The paper is organized as follows: The Background Study section reviews relevant background work, while Proposed System section introduces our proposed system and the machine learning models employed. Model evaluations and the discussion of results are presented next, leading to the Conclusion and Future Work section, which offers a conclusion that summarizes our findings and suggests directions for future research.

2 Background study

Over the past decade, there has been considerable exploration into the integration of ML into IDS [22, 23]. A review by Md.Alamgir at [24] employing various ensemble techniques shows that the suggested strategy utilizing the Random Forest technique beats previous strategies in terms of accuracy and FPR, usually above 99% with superior assessment metrics. The authors in [25] used an auto-encoder as a label. A deep-learning classification model [26] was trained on the KDD data set, which achieved an average accuracy of 85%.

In the study [27], A new architecture for deep neural networks that can train flexible and robust models for intrusion detection is established by merging the multichannel feature learning stage of the unsupervised setting with the supervised set of cross-channel feature correlation [28]. Specifically, in the unsupervised stage [29], two autoencoders are trained during the normal and attack procedures. The top layer of the decoder of these autoencoders revamps tests as a contribution to a similar space, permitting them to be utilized to create two different component vectors, allowing each organization stream to be addressed as a multichannel test [30, 31]. During the supervision phase, multichannel parameter convolution is utilized to understand how one channel affects the others [32, 33]. Since the examples come from two unmistakable distributions (typical and assault stream), the ordinary models should be more

pertinent to an assault than the portrayal remade utilizing a standard autoencoder, and vice versa [34]. Their expected dependency can improve the geographical removal of flows. The proposed neural network design can improve prediction accuracy in the following scenarios. On three benchmark data sets, they examined rival intrusion detection architectures [35, 36]. In the review [37], the PSO-Xgboost model is introduced because it outperforms competing models like Xgboost, Irregular Backwoods, Stowing, and Adaboost regarding overall order precision.

Imran *et. al.* [38], proposed a hybrid feature selection method based on the random forest model and Pearson correlation coefficient. For the machine learning (ML) model, the decision tree, AdaBoost, and K-nearest neighbor are trained and tested on the TON-IoT dataset. The dataset is fresh and includes features and attack kinds that are new and current [39]. Multilayer perceptrons (MLPs) and long short-term memory are trained and tested for deep learning [40, 41]. The criteria used for evaluation include recall, accuracy, and precision. Recent research has demonstrated that IDS with machine learning models can achieve high accuracy rates exceeding 90% [42, 43]. As a result, some studies have shifted their focus to comparative analyses, where numerous models are applied and assessed using varied datasets. Shema and Saad in 2023 [44] propose an innovative strategy employing a combination of deep learning and three-level algorithms to detect threats in IoT networks quickly and accurately. The suggested method is evaluated using the Bot-IoT dataset, and the outcomes demonstrate notable gains in detection performance over current techniques [45, 46]. The suggested method is a promising addition to the field of IoT security since it can be expanded to improve the security of additional IoT applications [47–53].

The paper [54] proposed a malware analysis and detection with machine learning algorithms to compute the difference in correlation symmetry (Naive Byes, SVM, J48, RF, and with the proposed approach) integrals could be used to detect harmful traffic on computer systems, thereby improving the security of computer networks. The findings demonstrated that DT (99%), CNN (98.76%), and SVM (96.41%) performed well in terms of detection accuracy when compared to other classifiers. In the paper [76], the authors proposed using adversarial machine learning for malware and intrusion detection scenarios. We concluded that, while their applicability in intrusion scenarios was not tested, a wide range of attacks were evaluated and shown to be successful in malware and intrusion detection. IDS's primary goal is to identify intrusion activity by looking for a suspicious trend in the events that are being recorded. The normal and attack user datasets were first gathered via the Internet of Things and fed into the system for training.

In light of this, a brand-new Bear Smell-based Random Forest (BSbRF) was created to detect intrusions accurately [55] by keeping an eye on their behavior about their threshold value. In the publication [56], the proposed traffic anomaly detection model is BAT. Bidirectional Long Short-Term Memory (BLSTM) and the attention mechanism are combined in the BAT model. The network flow vector comprised of packet vectors produced by the BLSTM model is screened using an attention mechanism to extract the essential characteristics needed to classify network traffic [57, 58]. in addition, a network intrusion detection technique combines deep hierarchical networks and hybrid sampling. First, we lower the noise samples in the majority category using one-side selection (OSS). Then, we utilize the Synthetic Minority Over-sampling Technique (SMOTE) [59] to augment the minority samples.

In [60], the authors proposed applying AI (Machine Learning) approaches for intrusion detection. They utilized the KDD dataset from the UCI archive. They executed different administered or supervised models to adjust non-arrangement algorithms for better execution. They have accomplished excellent outcomes in this work. To identify an adversary's attempts to inject undesired data into an IoT network, the authors in a study [61] created a lightweight

Table 1. Literature review comprising main studies.

Study	Methodology	Datasets	Key finding
Md.Alamgir [6]	ensemble techniques	various datasets	99 accuracy using RF
Yang Y [7]	auto-encoder	KDD	85 accuracy using deep learning
Andresini [8]	Un-sv + sv DNN	unspecified	better accuracy using benchmark datasets
Jiang H [9]	PSO-Xgboost	NSL-KDD	Outperformed competing models
Imran [10]	hybrid feature selection	TON-IoT	achieve high accuracy rates exceeding 90
Shema [11]	DL and three-level algorithms	Bot-IoT	improve the security of IoT applications
Martins [13]	adversarial machine learning	various datasets	successful in malware and intrusion
Alalayah [14]	Bear Smell-based RF	IoT-based	detect intrusions accurately
Su T [15]	BLSTM	benchmark	accuracy is largely improved
Jiang K [16]	hybrid sampling deep hierarchical network	NSL-KDD UNSW-NB15	accuracy can achieve 83 and 77
Nagaraja [17]	Machine Learning approaches	KDD	99% accuracy
Karatas [61]	supervised ML-based	CIC-IDS2018	Adaboost is the most successful
Zolanvari [19]	outline common IoT protocol	Built dataset	New types of attack such as backdoor, command injection
Yu Y, [20]	Few-Shot Learning (FSL)	NSL-KDD	92.34% accuracy
Xiao, [21]	CNN-IDS	KDD-CUP99	convert the original traffic vector into an image format
Kasongo [22]	feed-forward deep brain and a channel-based approach	NSL-KDD	made a precision of 86.62%
Lee J, [23]	AI-SIEM system	NSLKDD CICIDS2017)	capable of being employed as learning-based models)

https://doi.org/10.1371/journal.pone.0299666.t001

attack detection technique based on supervised machine learning-based support vector machines. The reenactment results showed that the recommended SVM-based classifier worked as far as grouping exactness and recognition time when empowered by a blend of a few complex elements.

The proposed work [62] outlined common IoT protocols and the vulnerabilities that go along with them. After that, a cyber-vulnerability assessment was conducted, which described how machine learning might be utilized to reduce these risks. Using merely 1% of the NSL-KDD KDDTrain+dataset for training, the intrusion detection approach described in the paper [63] achieved 92.34% accuracy for the KDD-Test and 85.75% accuracy for the KDD-Test-21. The review [64] proposed an organization interruption recognition method in light of a convolutional brain network-IDS (CNN-IDS). First, repetitive and unnecessary qualities in network traffic information are eliminated using different dimensionality decrease techniques. KDD-CUP99 dataset was used [65]. The results of AC, FAR, and timeliness reveal that the CNN–IDS model outperforms existing methods. In [66], a profound learning-based interruption recognition framework (IDS), because of feed-forward deep brain organizations (FFDNNs) and a channel-based highlight determination approach, was created. NSL-KDD dataset on the test set made a precision of 86.62%.

The authors in [67], proposed AI-SIEM system using various artificial neural network techniques, such as CNN, LSTM, and FCNN, together with event profiling for data preparation. The approach helped security analysts react quickly to cyber threats by focusing on differentiating between true positive and false positive signals. The authors of this work conducted all of the experiments using two real-world datasets and two benchmark datasets, NSLKDD and CICIDS2017. Five traditional machine-learning techniques (SVM, k-NN, RF, NB, and DT) were used to assess the performance comparison with current methodologies. As a result, the experimental findings of this work guarantee the applicability of our suggested techniques as learning-based models for network intrusion detection and demonstrate that while. Table 1 summarises the whole review.

3 Proposed system

In this part, we provide a cutting-edge intrusion detection algorithm that challenges the standard method, as shown in Fig 2. RNNs are used to train the data in our method. Random Forest is used to select features, and finally, a variety of classifiers, including Decision Trees, Random Forest, and Boosting algorithms (AdaBoost, boost, and XGBoost), are used. To assess the efficacy of this novel technique, it is applied to the NSL-KDD and CIC-2017 datasets. We execute crucial data pretreatment procedures to ensure the datasets are appropriate for analysis before applying our new technique. This comprises categorical feature encoding, resolving missing values, and data cleaning.

To improve the model's comprehension of complicated network intrusion scenarios, we train the data with RNNs to develop robust data representations. We go on to feature extraction after RNN data training. In this stage, we use the RNNs' learned information to extract pertinent characteristics from the data. This ensures that the model's knowledge of the subtleties and underlying patterns in the dataset informs the selection of the features. Using Random Forest as the feature selection method, we further improve the feature set after feature extraction. By doing feature selection after feature extraction, our methodology varies dramatically from traditional approaches. As a result, the model may direct the selection procedure using its learned data representations, potentially revealing non-trivial aspects that are significant to the context.

After classification, in terms of precision, recall, and F1 score, authors perform performance measures. Finally, the authors compare the work with existing ones. We use a variety of classifiers to perform intrusion detection using the chosen characteristics. Decision Trees (DT), Random Forest (RF), AdaBoost, Catboost, and XGBoost are some of the classifiers in this group. To test each classifier's capacity to discriminate between legitimate and malicious network activity, a dataset containing the chosen attributes is used to train each classifier. Our methodology is also designed to offer fast response times for real-time intrusion detection applications, ensuring timely decision-making.

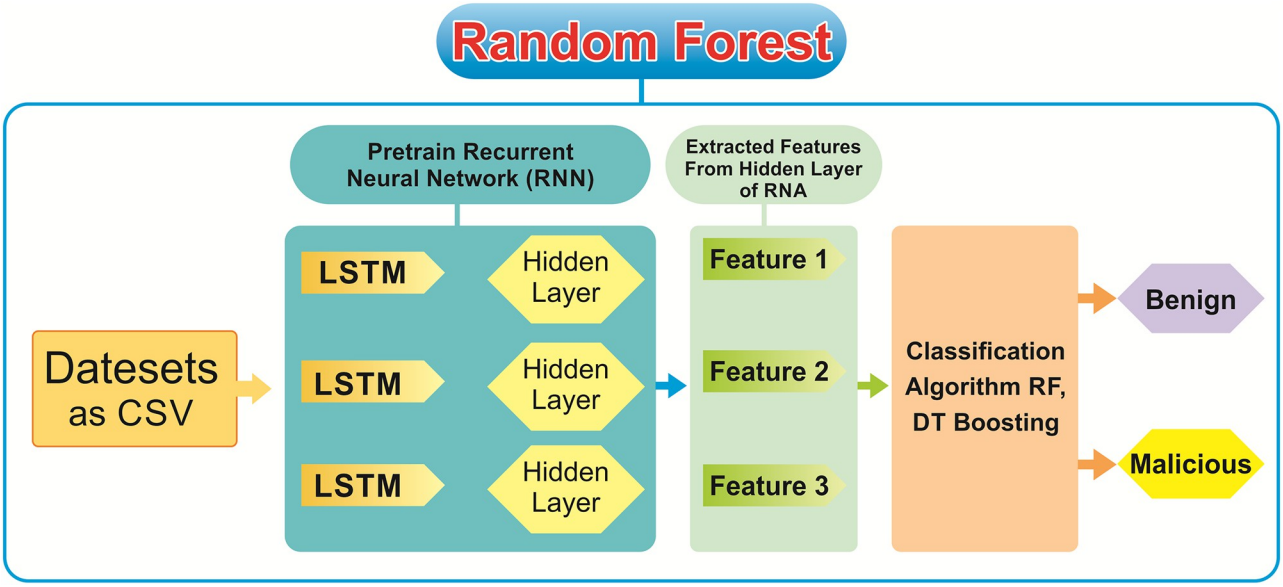

Fig 2. Proposed solution of IDS.

https://doi.org/10.1371/journal.pone.0299666.g002

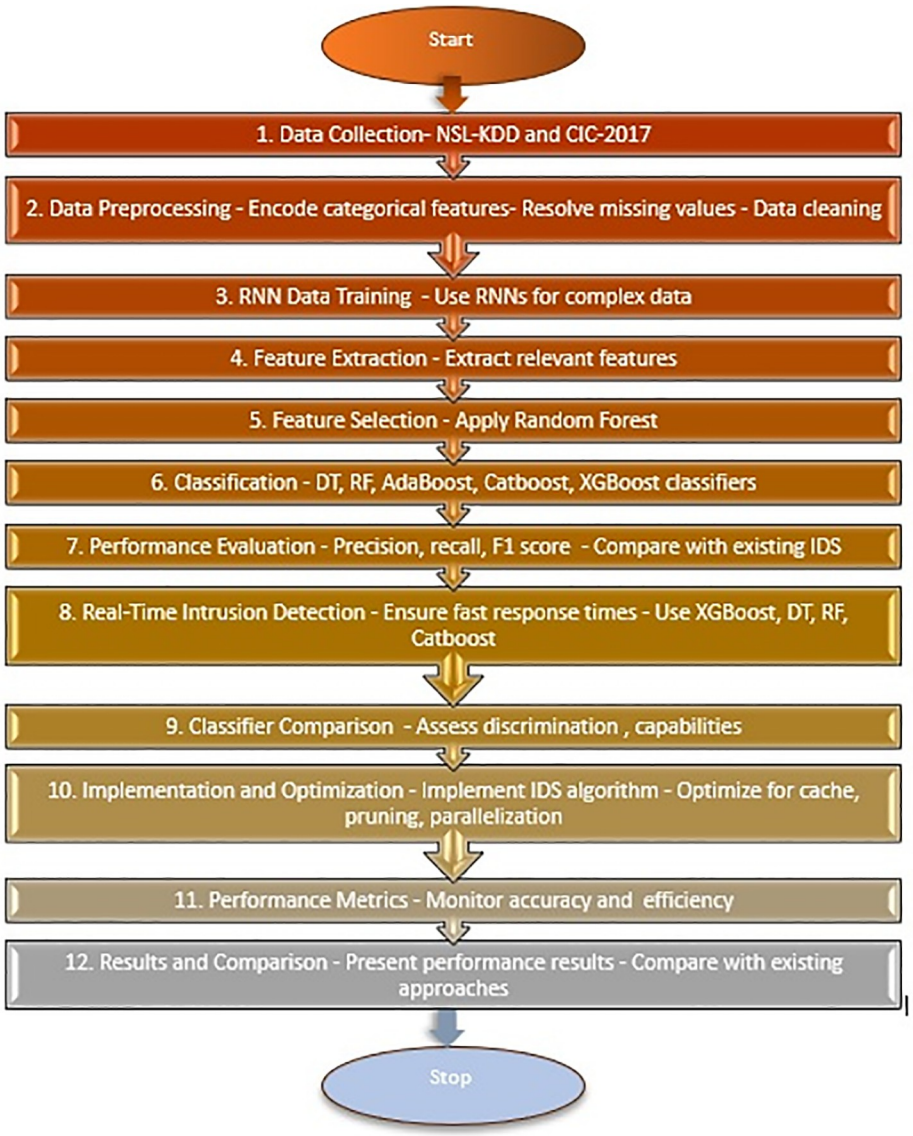

Fig 3. Flow chart of the proposed methodology.

https://doi.org/10.1371/journal.pone.0299666.g003

We use XG Boost, Decision Tree, Random Forest, and Catboost classifiers. XG Boost is considered one of the best machine learning algorithms because of cache optimization, auto pruning, parallelization, out-of-memory computation, regularization, and missing value handling. The five classifiers are used on the following datasets. The goal is to learn about data integrity and improve data prediction accuracy.

A flow chart of the proposed methodology is presented in Fig 3.

3.1 Dataset

The lack of accessible datasets is one of the main challenges facing researchers of intrusion detection systems (IDS). This is because many organizations are reluctant to share their network traffic data due to concerns about privacy and security. Nonetheless, having a reliable

dataset is essential for building an anomaly-based IDS and assessing its effectiveness. As a result, numerous datasets have been created by various organizations for research purposes. This section will introduce the datasets that were utilized in our research.

3.2 NSL-KDD dataset

NSL-KDD is a data collection proposed to address some of the shortcomings of the KDD'99 data set. The NSL-KDD train and test sets have a reasonable amount of recordings. This benefit makes it feasible to execute the trials on the entire collection rather than a small subset at random. The proportion of records chosen from each difficulty level group is inversely proportional to the proportion of records in the original KDD dataset. As a result, various machine learning approaches have a greater variety of categorization rates. This method improves the accuracy of assessing various learning strategies.

3.3 2017 CIC-IDS dataset

The 2017 CIC-IDS Dataset dataset is a well-known and widely used dataset that has gained popularity in recent years. It was developed in 2017 by the Canadian Institute for Cybersecurity (CIC) and offers a diverse range of operating systems, protocols, and attack types. The dataset is especially valuable as it covers a wide range of contemporary operating systems, including Windows, Mac OS, and Ubuntu, and was created in a comprehensive network environment with various components such as modems, firewalls, switches, and routers. Using protocols including HTTP, HTTPS, FTP, SHH, and other email protocols, the behaviour of 25 users was mimicked to provide benign traffic for the dataset. The most common assaults in 2016—brute force attacks, denial of service attacks, distributed denial of service attacks, intrusion attacks, Heartbleed attacks, botnet attacks, and port scan attacks—were recreated. Once the dataset was complete, it was made publicly available in CSV file format on the University of New Brunswick website [68]. The major drawback of the dataset is the high class imbalance issue, with over 70% of the traffic being benign and only a few attacks constituting less than 1% of it [69].

3.4 Data pre-processing

Data wrangling, also known as data cleaning, involves preparing data by removing unwanted values, null items, and other inconsistencies [70]. This process is a crucial stage in data mining, aimed at ensuring the data is clean for accurate analysis [71]. During this step, the authors apply preprocessing techniques to examine the data, removing any invalid or missing values, as well as duplicates. By eliminating absent or empty attributes and duplicates, the authors ensure the data is well-prepared for further analysis. This complex and essential step advances the clarity and precision of data analysis.

3.5 Data visualization

The graphic structure's data address it just through visual information on the information that it is feasible to choose [72]. Close perception information and Matplotlib, information depiction information is accessible. In this section, the authors exhibit our data set visually. It is enough to interpret the data better and make algorithm decisions. Figs 4 and 5 depict attacks in a graphical style that shows all intrusions. Fig 4 depicts attacks in a graphical style that shows all intrusions. The CIC-IDS2017 dataset comprises five categories, with the first one being benign (2096134), followed by DoS Hulk (172846), DDoS (128016), PortScan (90819), and finally DoS GoldenEye (10286). In the NSL-KDD dataset, the first one is normal (67343),

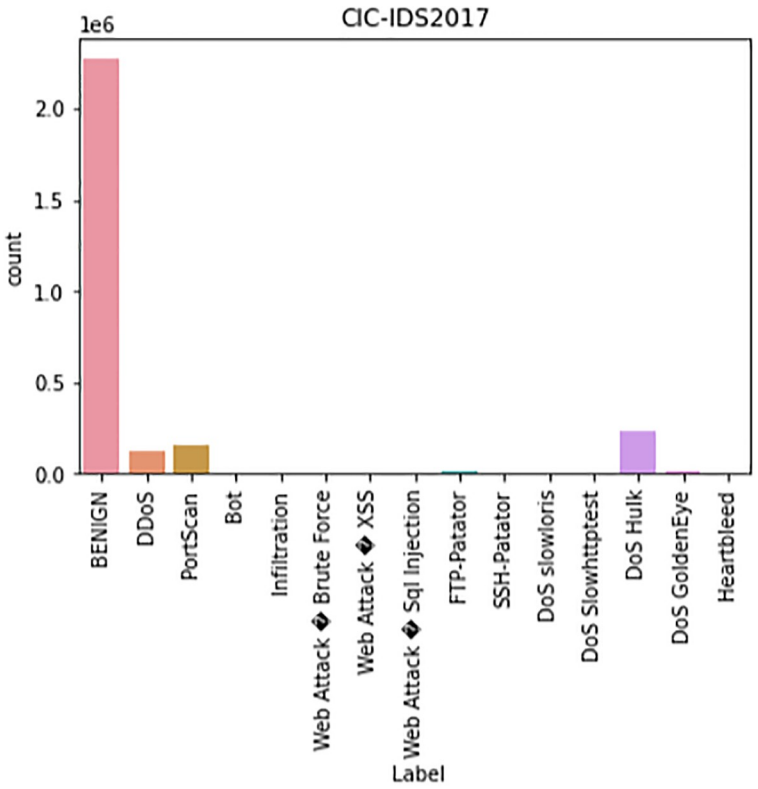

Fig 4. Depicts attacks on CIC-IDS2017.

https://doi.org/10.1371/journal.pone.0299666.g004

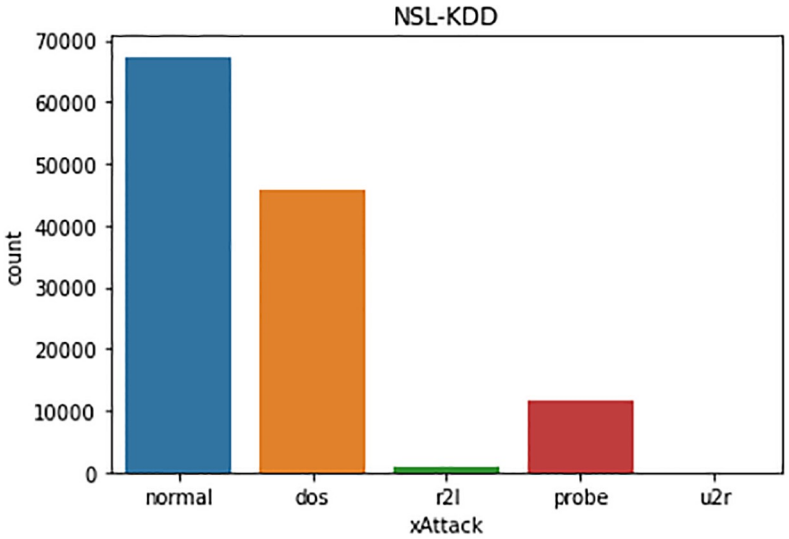

Fig 5. Depicts attacks on NSL-KDD.

https://doi.org/10.1371/journal.pone.0299666.g005

followed by Dos (45929), Probe (11656), r2l (995), and u2r(52). Now, we use the above actual intrusion for classification and prediction by our understudy machine learning models. It is our target class for a model. The next step is featuring scaling to remove overfitting from the data set to make the data have a normal form. That can help in model training for best learning.

All the machine learning algorithms use input data [73]. This input data set has structural columns as features [74]. All calculations need the utilization of information highlights with specific characters to work appropriately. Providing an input data collection compatible with the machine learning model specifications is the primary aim of feature engineering. Accordingly, the authors start by changing overall downright properties into mathematical qualities. Second, to work on the AI (Machine Learning) model exhibition. Authors use the feature scaling technique to balance the independent variables in the dataset over a specific range. It handles the value, magnitude, value, or unit of height variation during data processing. Without feature scaling, regardless of the unit of value, the machine learning algorithm will give less importance to smaller values and more importance to higher values. For feature scaling implementation, two optimal approaches given below are used.

3.6 Standardization

As a first approach, standardization is utilized, and it is the method involved with deducting your information by the process for all perceptions and partitioning the outcome by the standard deviation, followed by scaling your perceptions [75]. In AI, the accompanying recipe is utilized for normalization. It is a very effective strategy for rescaling the information to create a dispersion with a mean of nothing and a change of one.

$$Xnew = \frac{x_i - Xmean}{standardDeviation} \tag{1}$$

3.7 Normalization

Normalization includes deducting your perception from the base, everything being equal, separating by the most significant number of perceptions, and afterward scaling the component. This approach rescales elements or perceptions with a circulation esteem between 0 and 1.

$$Xnew = \frac{x_i - min(x)}{man(x) - min(x)} \tag{2}$$

The authors apply a typical scalar feature scaling approach for feature scaling in our proposed study. Since it is the most effective method for scaling different label features in supervised learning models. Machine learning algorithms utilize artificial intelligence to take and refine data without being expressly modified. AI is worried about the advancement of PC frameworks that can acknowledge and gain from data [76]. Prior knowledge is used to supervise classification algorithms that employ categorical data sets for classification and prediction tasks. Also, the authors used the model optimization technique to improve model performance. The following are the results of the algorithms used in this research study.

4 Machine learning models

Machine learning (ML) and its potential applications in numerous fields have drawn increasing attention in recent years [77]. The ability of ML models to predict or categorize data is very strong. In Intrusion Detection Systems (IDS), ML is used to distinguish between legitimate

and malicious network traffic. Some of the most popular ML models in IDS are briefly described in this section.

4.1 Recurrent Neural Network (RNNs)

Recurrent neural networks (RNNs) are a type of artificial neural network designed for processing data sequences. The fundamental structure of an RNN involves the following components. Input Sequence: An RNN receives a data sequence as input. This sequence can be of various lengths and represent time series data, natural language phrases, or any other sequential data. let X_t be the input at time step t,W_{hx} is the weight matrix connecting the hidden state to the input. H_{t-1} is the previous hidden state, W_{hh} is the weight of the matrix for recurrent connections H_t be the hidden state at the time step t. Then, the combination of the hidden state can be expressed as

$$X_t = activation(W_{hx}.x_t) + W_{hh}.H_{t-}1 \tag{3}$$

$$(x) = \frac{1}{1+e^z} \tag{4}$$

Rectified Linear unit (ReLU): ReLU(x) = max (0, x) These functions are applied element-wise to the outputs of matrix multiplication.

4.2 Random Forest

An RF is a troupe learning strategy that frames an enormous gathering of free-D-tree and results in the method of the name forecasts of the multitude of trees. This technique has more prominent registering costs yet decreases overfitting of the information, which happens when a model adjusts too stringently to a specific arrangement of information, having unfortunate capacities for speculation. An RF classifier depends on a group of RF classifiers H(x|θ1. . .. . .h(x|) based on a classification tree with parameter θk is randomly selected the random vector model θ For the last classification, f(x), which is used to combine the classifiers {$h_x(x)$} *respectively*.

4.3 Decision Tree

One of the fundamental Supervised ML techniques, DT applies a series of decisions (rules) to characterize the dataset and perform regression on it. It has a predefined objective variable and is upheld by a leaf hub. A portion of the leaf hubs is wiped out from the D-tree to avoid Overfitting. Entropy and data gain ought to be determined for the D-tree, so how entropy ought to be selected is given in Eq 5.

$$E(S) = \sum_{x \propto X}^{n} - p(a)E(a) \tag{5}$$

S is a dataset, X is a collection of classes inside S, and p is a ratio of the number of entries within class X. Eq 6 thus illustrates the calculation of knowledge gain

$$Gain(A,S) = E(S) - \sum_{x \propto T}^{n} - p(a)E(a) \tag{6}$$

Where subsets are created from the dataset S as a T.

4.4 AdaBoost classifier

Machine learning is also used as a classification algorithm that uses labelled data. It is the same as XGBoost but utilizes only two levels of the decision tree. It works on probability. This approach is used to represent the classification performance of machine learning. Authors may acquire an improved understanding of the classification model's accuracy and the sort of error generated by computing the confusion matrix. It is used to measure classification accuracy and categorize accurate and predicted labels. They present a visual representation of the classifier's performance.

4.5 XGBoost classifier

XgBoost is another machine learning algorithm that is analyzed in this research. XGBoost is a widely used machine learning technique, and it's gotten a lot of attention recently since it's speedy, especially when working with enormous data sets. The accuracy, speed, and size of XGBoost have matured into models for data science challenges. As a result, it is critical to achieve high levels of accuracy and execution speed. XGBoost Classifier is 100 times quicker than previous algorithms and similar to a tree. It functions admirably with enormous informational indexes [78].

4.6 CatBoost

Below are the results and performance obtained through the CatBoost classifier, a gradient-boosting algorithm working on a tree. It is compelling and widely used for future forecasting for probability. This model can perform at high performance without a GPU. This means this model is more likely to perform than other boosting models.

5 Model evaluation

This section includes results from classifiers and an overview of our proposed model. For the intended study, the authors chose the Python programming language. This programming language is simple and can be customizable. The language's best-desired position is to employ minimal code and implicit libraries with explicit practicality. In the Jupyter Notebook, authors use Python. The Jupyter Notebook is a fast, open-source editor. The authors selected the CIC-IDS2017 and NSL-KDD, containing information about intrusion data in the latest version. The authors utilise the most recent dataset to categorize the current level of incursion. The total number of rows and columns in the CIC-IDS2017 dataset is 2830743 and 79, respectively. In NSL-KDD, there are (125973) rows and 42 columns. The authors used essential Python libraries to read and write data for further processing. Each of these libraries has its capabilities for doing various tasks.

5.1 Feature selection CIC-IDS

A crucial stage in creating a machine learning model is feature selection. By focusing on the features that have the greatest impact on the target variable, feature selection reduces the complexity of the dataset. By doing so, it can help in improving the accuracy of the model, reducing the training time, and avoiding overfitting. In paper [79], the feature selection problem was turned into an optimization problem using the Hybrid Ant-Bee Colony Optimisation (HABCO) technique. In [80], the fusion of the Statistical Importance-based feature selection technique is used. The CIC-IDS2017 dataset is used in its entirety and contains about 80 features. Training the model without any feature selection will take time. In addition, some features could introduce noise and decrease the model's accuracy. Following RNN-based pre-

Table 2. Accuracy results on the CIC-IDS2017 dataset.

Model	Accuracy	F1-Score	Precision	Recall
DecisionTree	0.9987	1.00	1.00	1.00
height *RandomForest*	0.9989	1.00	1.00	1.00
XGBoost	0.9716	0.94	0.97	0.92
AdaBoost	0.9958	0.99	0.99	0.99
CatBoost	0.9993	1.00	1.00	1.00

https://doi.org/10.1371/journal.pone.0299666.t002

processing, we use Random Forest as a feature selection technique. This hybrid method takes advantage of RNNs' temporal modeling skills as well as Random Forest's capacity to rank and choose the most important characteristics.

5.2 Performance evaluation

The confusion matrix represents the total number of actual and expected classification labels. It combines True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP) values to produce actual and anticipated labels. We calculate the model's classification and prediction accuracy based on these values:

1. True-Negative: the number of accurate predictions that a case has received.
2. True-Positive: the percentage of accurate predictions indicating a positive outcome
3. False-Positive: the number of inaccurate predictions that show a positive instance.
4. False-Negative: the quantity of false forecasts that lead to a negative event.

The labels of the confusion matrix are false negative, true positive, true negative, and false positive. Furthermore, using the aforementioned confusion matrix, we assess the performance of the suggested model.

5.3 Performance measure of CIC-IDS

In this step, the reliability of the models is achieved. Table 2 displays the accuracy score for each model, indicating that all models achieved an accuracy score higher than 99%. Moreover, the standard deviation of each model's accuracy is minimal, demonstrating that, using the CIC-IDS2017 dataset, the accuracy of each model is quite consistent. On the other hand, the CatBoost model performed the best, achieving an accuracy of 99.99%, which is a significant accomplishment. Table 3 compare the accuracy of the proposed approach with different models.

We assessed the accuracy of the classification outcomes and predictive findings using the confusion matrix. Figs 6–10 display the confusion matrices and classification results for Decision Tree, Random Forest, Naïve Bayes, XGBoost, AdaBoost, and CatBoost.

Table 3. Using the CIC dataset, accuracy comparison with other literature.

Model	Proposed Model	Tuan-Hong Chua [22]	Kostas [30]
DecisionTree	1.00	0.995	0.95
RandomForest	1.00	0.996	0.94
XGBoost	0.9716	-	-
AdaBoost	0.9958	-	-
CatBoos	0.9993	-	-

https://doi.org/10.1371/journal.pone.0299666.t003

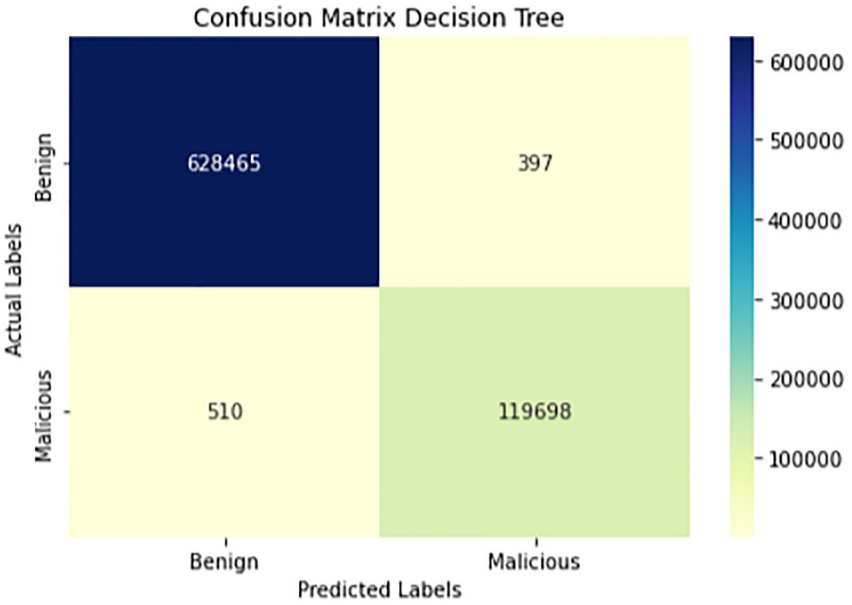

Fig 6. Confusion matrix of DT.

https://doi.org/10.1371/journal.pone.0299666.g006

5.4 Model evaluation NSL-KDD

In this work, we employed a variety of machine learning methods, including Decision Trees (DT), Random Forest (RF), XGBoost, CatBoost, and AdaBoost. We achieved an outstanding 99% accuracy on the NSL dataset for intrusion detection. This exceptional precision underscores the effectiveness of our strategy in enhancing cybersecurity measures. Our findings

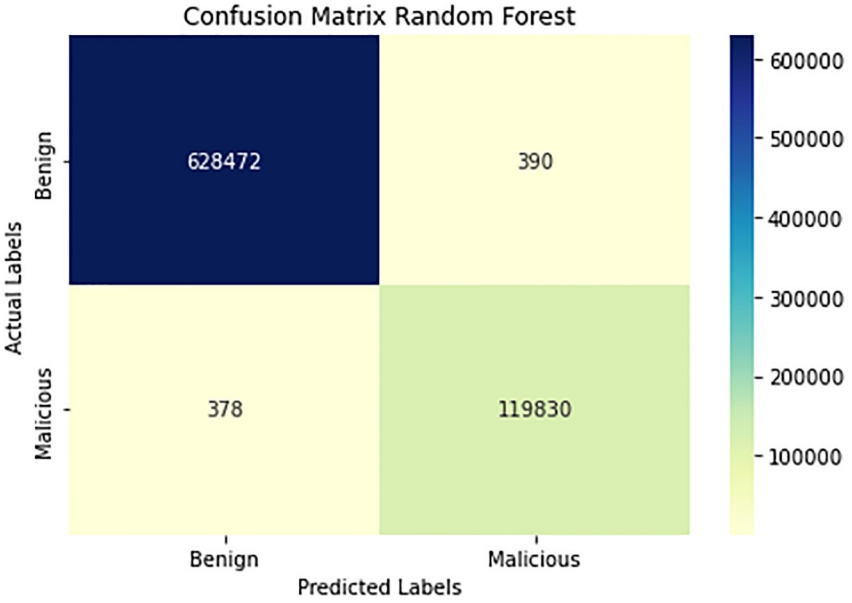

Fig 7. Confusion matrix of RF.

https://doi.org/10.1371/journal.pone.0299666.g007

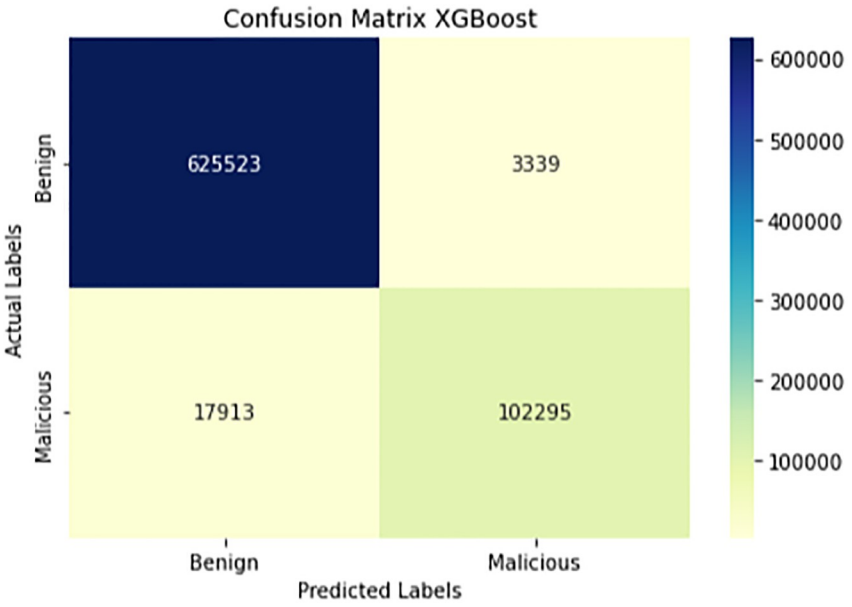

Fig 8. Confusion matrix of XGboost.

https://doi.org/10.1371/journal.pone.0299666.g008

highlight the considerable potential of these algorithms for intrusion detection tasks on the NSL dataset as shown in Table 4, setting the stage for future efforts in cybersecurity.

Confusion matrix and classification results of the NSL-KDD dataset are displayed in Figs 11–15. Figures expound the confusion matrix for all model execution recognizable proof. On the NSL-KDD dataset, the results of DT, RF, XGBoost, and CatBoost are consistent with those obtained on the CIC-IDS2017 datasets. However, the XGBoost algorithm achieved an excellent

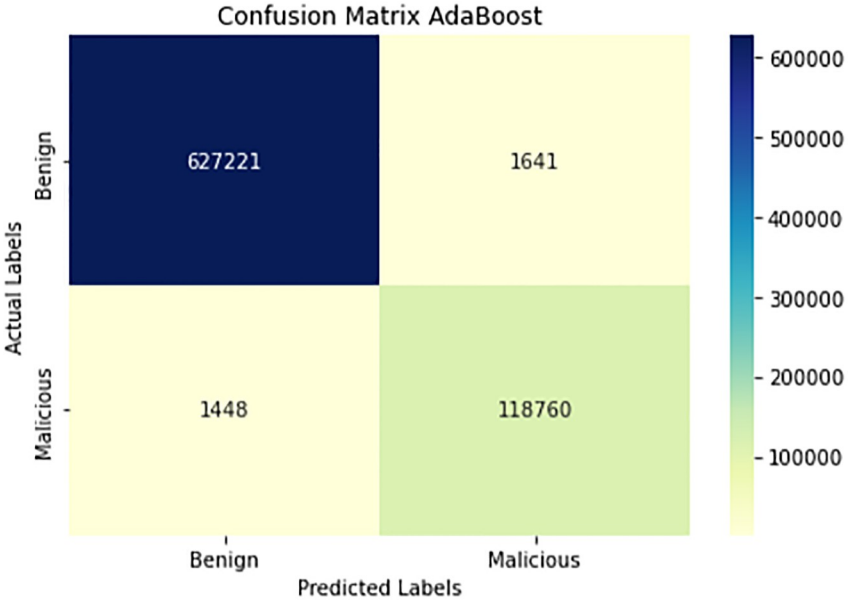

Fig 9. Confusuion matrix of AdaBoost.

https://doi.org/10.1371/journal.pone.0299666.g009

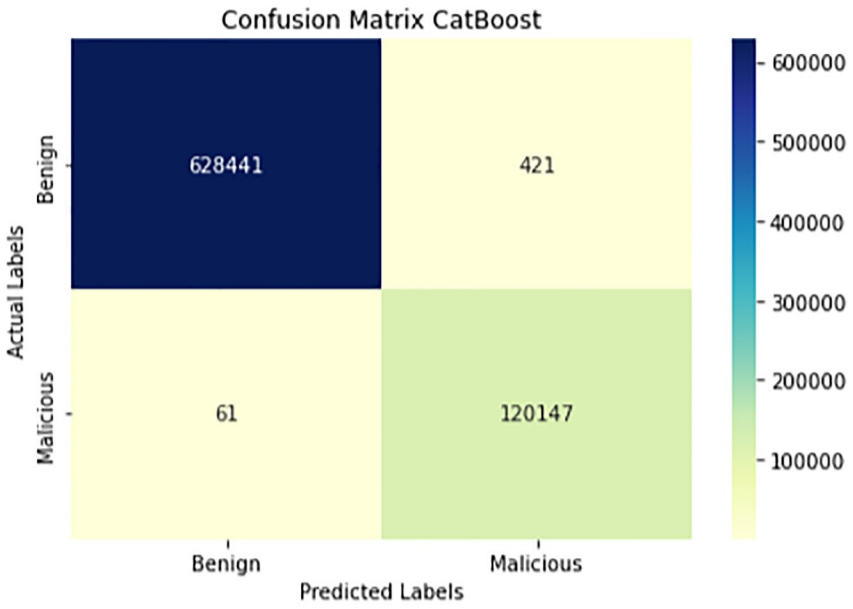

Fig 10. Confusion matrix of CataBoost.

https://doi.org/10.1371/journal.pone.0299666.g010

accuracy of 99.97% on the NSL dataset, demonstrating its extraordinary ability in intrusion detection. Table 5 shows the comparison analysis of the proposed model with other models present in the literature.

Additionally our approach of training the model before classification not only enhances accuracy but also optimizes the computational efficiency of intrusion detection. model performance comparisons on the training and testing dataset are shown in Figs 16–18. In Fig 16 our approach of training the model before feature selection leads to significantly better results compared to the traditional approach, Furthermore, relative to their recall scores, both models have inferior precision scores on the benign class of the *NSLKDD* and CIC-2017 datasets. According to Figs 17 and 18, AdaBoost are the most time-consuming models for training, while DT, RF, CatBoost, and XGboost remain the most efficient

The time complexity of each model is a significant component that impacts the training duration in addition to the previously listed factors. Each model has its unique algorithmic structure and computational requirements, which result in different time complexities. For instance, AdaBoost typically requires more computations and memory compared to DT, RF, and Catboost due to their complex algorithms. Consequently, AdaBoost is more time-

Table 4. The accuracy result on NSL-KDD dataset.

Model	Accuracy	F1-Score	Precision	Recall
DecisionTree	0.9965	1.00	1.00	1.00
RandomForest	0.9969	1.00	1.00	1.00
XGBoost	0.9976	1.00	1.00	1.00
AdaBoost	0.9846	0.98	0.97	0.98
CatBoost	0.9960	0.99	1.00	0.99

https://doi.org/10.1371/journal.pone.0299666.t004

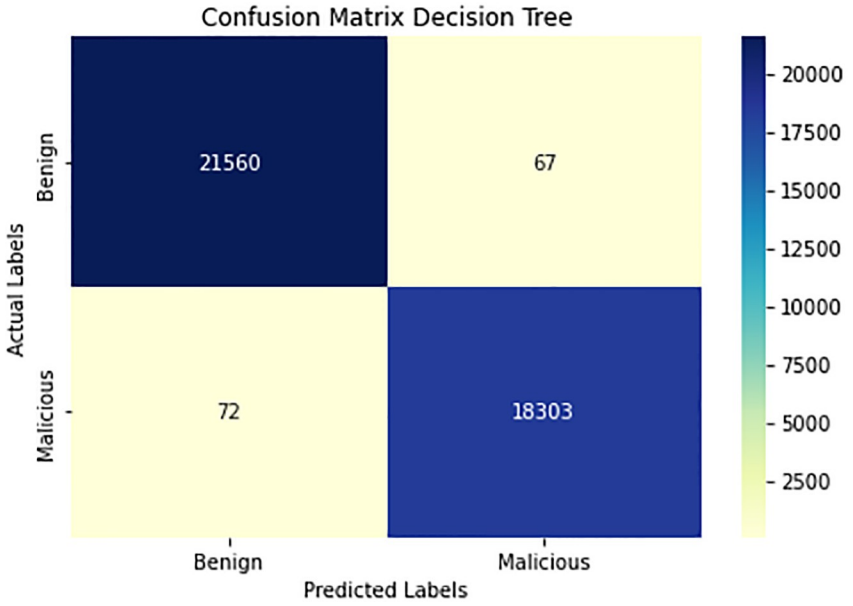

Fig 11. Confusion matrix of DT.

https://doi.org/10.1371/journal.pone.0299666.g011

consuming for training, whereas DT, RF, NB, XGboost, and CatBoost are less demanding and, therefore, more efficient in terms of training time.

6 Results and discussion

In our method, we present the outcomes of our innovative approach, which entails training the data using recurrent neural networks (RNNs) before feature selection with random forests

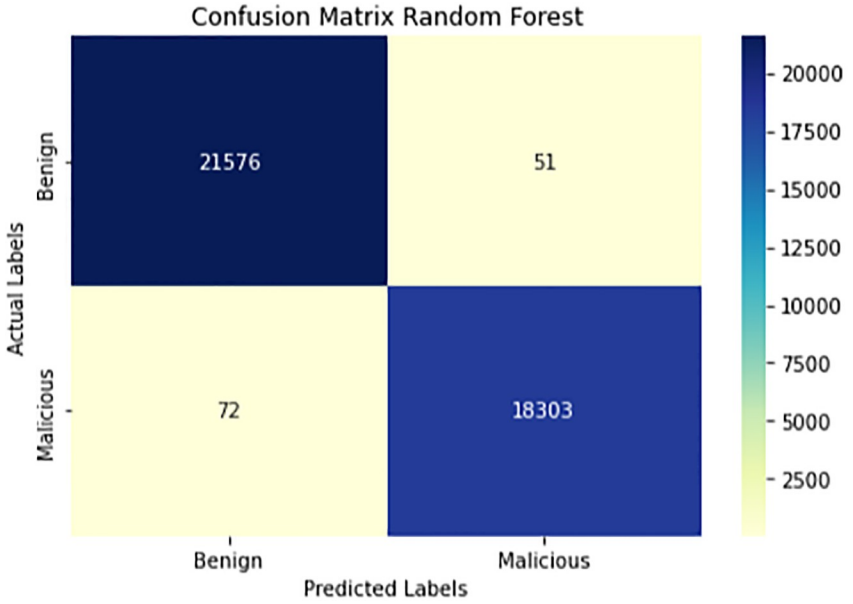

Fig 12. Confusion matrix of RF.

https://doi.org/10.1371/journal.pone.0299666.g012

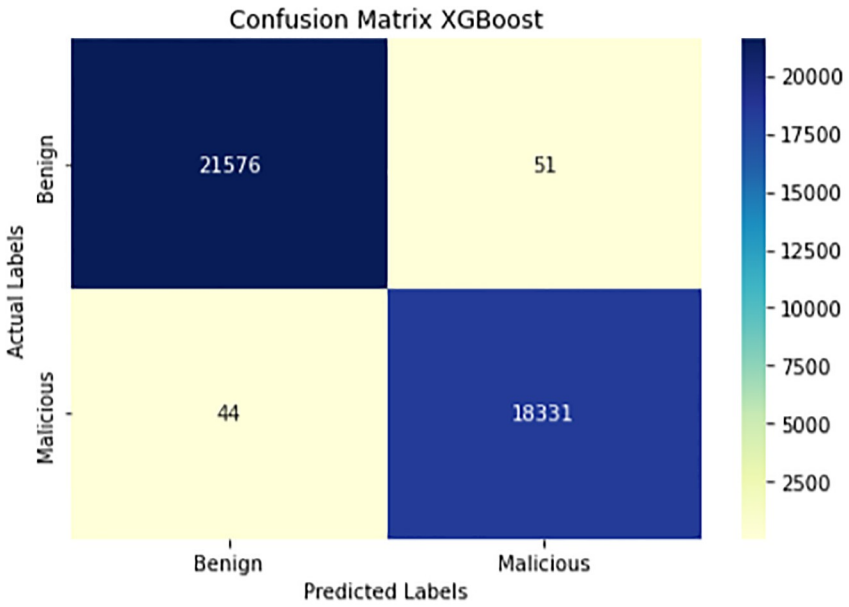

Fig 13. Confusion matrix of XGboost.

https://doi.org/10.1371/journal.pone.0299666.g013

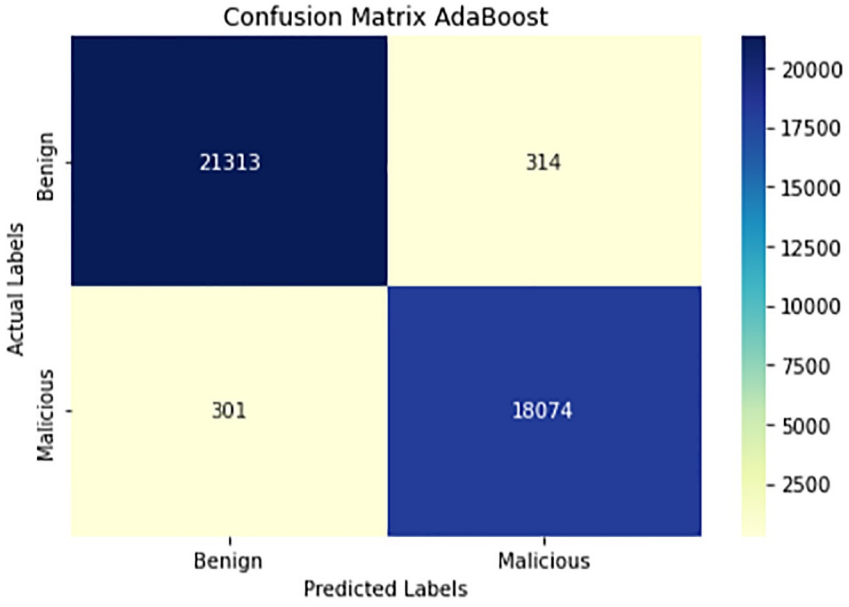

Fig 14. Confusuion matrix of AdaBoost.

https://doi.org/10.1371/journal.pone.0299666.g014

(RF), followed by classification using different classifiers, including decision trees (DT), random forests (RF), and boosting algorithms (AdaBoost and XGBoost).

In this study, we introduce a novel approach for intrusion detection that initially employs Recurrent Neural Networks (RNNs) for data training, followed by Random Forest for feature selection, and various classifiers for the classification process. The experimental results on the

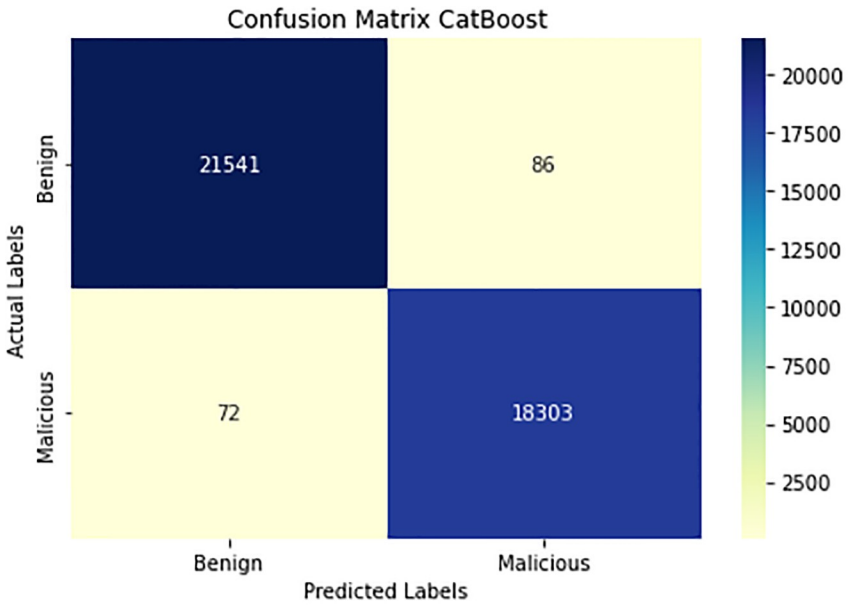

Fig 15. Confusuion matrix of CataBoost.

https://doi.org/10.1371/journal.pone.0299666.g015

Table 5. Using the NSL dataset to compare accuracy to other research.

Model	Proposed Model	Tuan-Hong Chua [22]	Kostas [30]
DecisionTree	1.00	0.9994	-
RandomForest	1.00	0.9994	-
XGBoost	1.00	-	-
AdaBoost	0.98	-	-
CatBoost	1.00	-	-

https://doi.org/10.1371/journal.pone.0299666.t005

NSL-KDD and CIC-2017 datasets as shown in Tables 6 and 7 illustrate the effectiveness of our strategy in achieving high accuracy while reducing training time. Our method combines the benefits of enhanced accuracy and computational efficiency, making it a practical choice for real-world intrusion detection systems. Our research challenges the traditional sequence of operations and highlights the importance of data-driven feature selection following model training, thereby advancing the field of intrusion detection. We believe our approach opens new avenues for efficient and accurate intrusion detection and warrants further investigation in practical application settings. Our focus lies on the unique aspect of our strategy—utilizing Recurrent Neural Networks (RNNs) for initial model training before employing Random Forest for feature selection. The aim of this hybrid technique is to boost the model's performance.

In this work, we conducted two tests: one using the datasets from CIC and the other using the dataset from NSL-KDD. Interestingly, the outcomes of both experiments were the same. Our research indicates that XGBoost and CatBoost are the most suitable models for environments where the system's infrastructure is regularly updated and the cost of cyberattacks is significant. This is because XGBoost and CatBoost balance prediction time and resistance to overfitting effectively. Conversely, if a system's infrastructure is not frequently updated and is less vulnerable to large-scale attacks, Decision Trees (DT) and Random Forest (RF) are

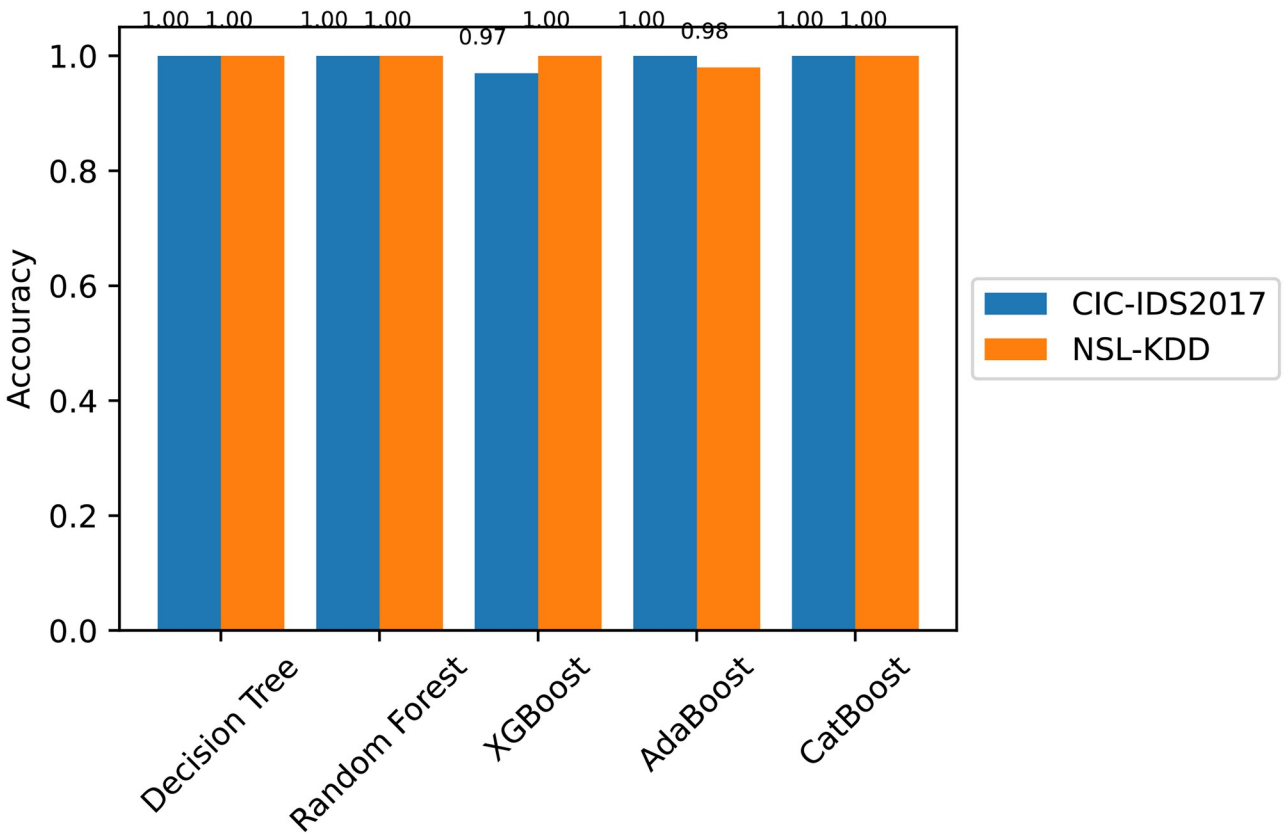

Fig 16. Models' F1-score and accuracy on the NSL-KDD dataset and CIC207.

https://doi.org/10.1371/journal.pone.0299666.g016

preferable options. DT provides the best accuracy on the training dataset in both studies, proving to be very effective in training and classifying new samples.

7 Conclusion and future work

In this study, we aimed to address the critical issue of reaction time efficiency while advancing the accuracy of intrusion detection. Our approach involves a two-step process: initial model training using Recurrent Neural Networks (RNNs), followed by feature selection with Random Forest, which has demonstrated outstanding results on two well-known datasets, NSL and CIC-2017. For our experiments, we selected these datasets to test five machine learning models: XGBoost, CatBoost, Random Forest, Decision Tree, and AdaBoost. We standardized the data before applying the suggested supervised learning algorithms to generate classification and prediction results.

Our method stands out because it strikes a balance between accuracy and reaction time, offering a practical solution to the intricate issue of intrusion detection. We employ Recurrent Neural Networks (RNNs) for initial model training, enhancing the system's resilience and adaptability. Following this, Random Forest is utilized for feature selection, ensuring the model's effectiveness. Our findings carry significant implications for the cybersecurity industry, marking a considerable step forward in bolstering an organization's security posture. This enables quicker and more accurate threat assessments, which we believe will greatly enhance incident response capabilities in real-world cyber environments. In our view, the combination

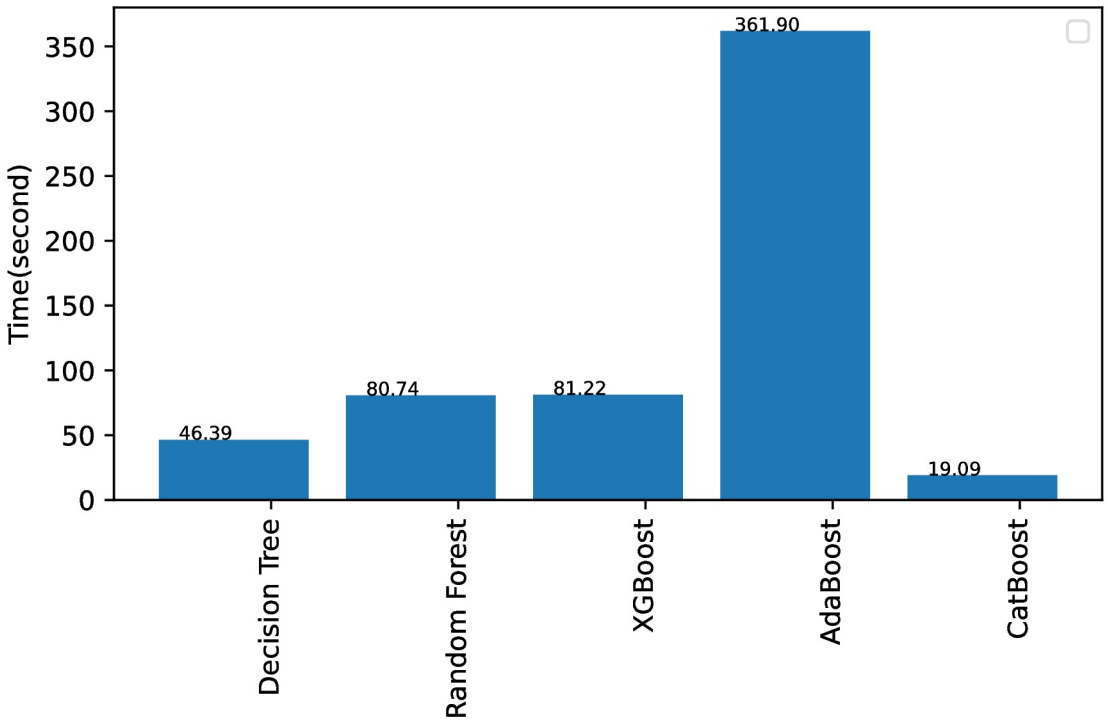

Fig 17. CIC-2017 dataset training period.

https://doi.org/10.1371/journal.pone.0299666.g017

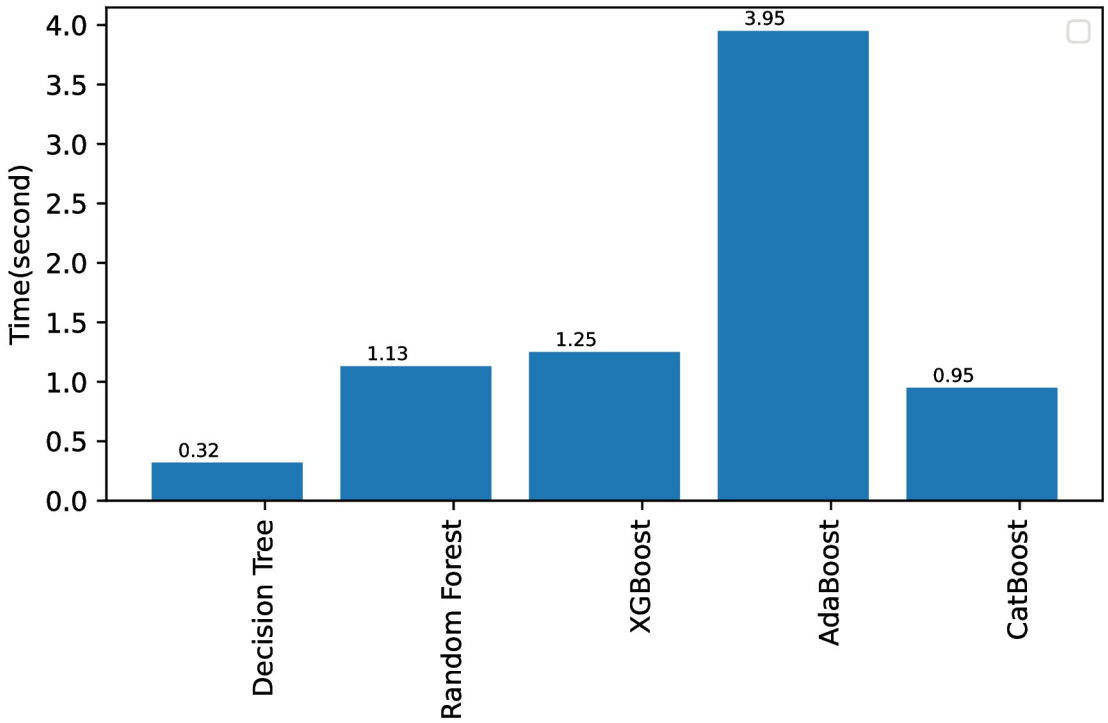

Fig 18. NSLKDD dataset training period.

https://doi.org/10.1371/journal.pone.0299666.g018

Table 6. NSL-KDD dataset results.

Model	Accuracy	F1-Score	Precision	Recall	Training time in (s)
RNN + RF + DT	0.9965	1.00	1.00	1.00	0.32
RNN + RF + RF	0.9969	1.00	1.00	1.00	1.13
RNN + RF + XGboost	0.9976	1.00	1.00	1.00	1.25
RNN + RF + Adaboost	0.9846	0.98	0.97	0.98	3.95
RNN + RF + Catboost	0.9960	0.99	1.00	0.99	0.95

https://doi.org/10.1371/journal.pone.0299666.t006

Table 7. CIC-IDS2017 result.

Model	Accuracy	F1-Score	Precision	Recall	Training time in (s)
RNN + RF + DT	0.9987	1.00	1.00	1.00	46.39
RNN + RF + RF	0.9989	1.00	1.00	1.00	80.74
RNN + RF + RF	0.9716	0.94	0.97	0.92	81.22
RNN + RF + Adaboost	0.9958	0.99	0.99	0.99	361.90
RNN + RF + Catboost	0.9993	1.00	1.00	1.00	19.09

https://doi.org/10.1371/journal.pone.0299666.t007

of high accuracy and rapid reaction times is crucial for improving incident response in actual cybersecurity settings.

Although our method has shown remarkable promise, we recognize several limitations, including the necessity for further testing across diverse datasets and the risk of over-fitting specific datasets. Future research should explore how effectively our method can scale to accommodate larger datasets, more complex infiltration scenarios, and evolving cybersecurity threats. In conclusion, our research marks a significant advancement in the field of intrusion detection. We have demonstrated that achieving high accuracy does not have to compromise reaction time effectiveness. Our innovative approach paves the way for enhancing the practical efficacy of intrusion detection systems. Our findings offer valuable insights for developing more efficient and reliable cybersecurity solutions as the cybersecurity landscape continues to evolve.

NSL-KDD and CIC-2017 were the only datasets utilized in the study. While these datasets are widely recognized benchmarks in intrusion detection research, they may not fully capture the diversity of real-world situations and network configurations. Employing a broader range of datasets could enhance understanding of our strategy's generalizability. Additionally, class imbalance, a common issue in intrusion detection datasets where some classes of intrusions are significantly less common than others, can lead to model bias towards the majority class, negatively impacting the performance of minority classes. Improving the model's resilience by addressing this imbalance through methods such as oversampling, undersampling, or employing sophisticated algorithms designed specifically for imbalanced data could mitigate this issue.

Author Contributions

Conceptualization: Nasrullah Khan, Sadaqat ur Rehman.

Data curation: Nasrullah Khan, Zahid Khan.

Formal analysis: Nasrullah Khan, Muhammad Ismail Mohmand, Zia Ullah, Wadii Boulila.

Funding acquisition: Sadaqat ur Rehman.

Investigation: Nasrullah Khan, Zahid Khan.

Methodology: Nasrullah Khan, Sadaqat ur Rehman.

Project administration: Muhammad Ismail Mohmand, Sadaqat ur Rehman, Wadii Boulila.

Resources: Nasrullah Khan.

Software: Nasrullah Khan, Sadaqat ur Rehman.

Supervision: Sadaqat ur Rehman.

Validation: Zia Ullah.

Writing – original draft: Nasrullah Khan.

Writing – review & editing: Muhammad Ismail Mohmand, Sadaqat ur Rehman, Zia Ullah, Zahid Khan, Wadii Boulila.

References

1. Yu Jiadiet al. "An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing". *IEEE Transactions on Mobile Computing* 20.2 (2019): 337–351. https://doi.org/10.1109/TMC.2019.2947468
2. Jiang Yunhao, and Li Xin. "Broadband cancellation method in an adaptive co-site interference cancellation system". *International Journal of Electronics* 109.5 (2022): 854–874. https://doi.org/10.1080/00207217.2021.1941295
3. Awajan Albara. "A novel deep learning-based intrusion detection system for IOT networks". *Computers* 12.2 (2023): 34. https://doi.org/10.3390/computers12020034
4. Yin Y., Guo Y., Su Q., & Wang Z. (2022). Task Allocation of Multiple Unmanned Aerial Vehicles Based on Deep Transfer Reinforcement Learning. *Drones*, 6(8), 215. https://doi.org/10.3390/drones6080215
5. Chen J., Xu M., Xu W., Li D., Peng W., Xu H. (2023). A Flow Feedback Traffic Prediction Based on Visual Quantified Features. *IEEE Transactions on Intelligent Transportation Systems*, 24(9), 10067–10075. https://doi.org/10.1109/TITS.2023.3269794
6. Ding Y., Zhang W., Zhou X., Liao Q., Luo Q., Ni L. M. (2021). FraudTrip: Taxi Fraudulent Trip Detection From Corresponding Trajectories. *IEEE Internet of Things Journal*, 8(16), 12505–12517. https://doi.org/10.1109/JIOT.2020.3019398
7. Li Hongboet al. "Public-key authenticated encryption with keyword search supporting constant trapdoor generation and fast search". *IEEE Transactions on Information Forensics and Security* 18 (2022): 396–410. https://doi.org/10.1109/TIFS.2022.3224308
8. Teodorescu Cosmin Alexandru. "Perspectives and reviews in the development and evolution of the zero-day attacks". *Informatica Economica* 26.2 (2022): 46–56. https://doi.org/10.24818/issn14531305/26.2.2022.05
9. Zheng, Wenfeng et al. "PAL-BERT: An Improved Question Answering Model".
10. Chen J., Wang Q., Peng W., Xu H., Li X., Xu W. (2022). Disparity-Based Multiscale Fusion Network for Transportation Detection. *IEEE Transactions on Intelligent Transportation Systems*, 23(10), 18855–18863. https://doi.org/10.1109/TITS.2022.3161977
11. Li S., Chen J., Peng W., Shi X., & Bu W. (2023). A Vehicle Detection Method Based on Disparity Segmentation. *Multimedia Tools and Applications*, 82(13), 19643–19655. https://doi.org/10.1007/s11042-023-14360-x
12. Xu X., Liu W., & Yu L. (2022). Trajectory Prediction for Heterogeneous Traffic-Agents Using Knowledge Correction Data-Driven Model. *Information Sciences*, 608, 375–391. https://doi.org/10.1016/j.ins.2022.06.073
13. Cheng Boet al. "Situation-aware dynamic service coordination in an IoT environment". *IEEE/ACM Transactions On Networking* 25.4 (2017): 2082–2095. https://doi.org/10.1109/TNET.2017.2705239
14. Chua, Tuan-Hong and Salam, Iftekhar. "Evaluation of machine learning algorithms in network-based intrusion detection system". *arXiv preprint arXiv:2203.05232* (2022).
15. Li Xuetao, and Sun Yi. "Stock intelligent investment strategy based on support vector machine parameter optimization algorithm". *Neural Computing and Applications* 32 (2020): 1765–1775. https://doi.org/10.1007/s00521-019-04566-2

16. Li Xuetao, and Sun Yi. "Application of RBF neural network optimal segmentation algorithm in credit rating". *Neural Computing and Applications* 33 (2021): 8227–8235. https://doi.org/10.1007/s00521-020-04958-9

17. Yan A., Liu R., Cui J., Ni T., Girard P., Wen X., et al. (2023). Designs of BCD Adder Based on Excess-3 Code in Quantum-Dot Cellular Automata. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 70(6), 2256–2260. https://doi.org/10.1109/TCSII.2023.3237695

18. Sharafaldin, Iman, Lashkari, Arash Habibi, and Ghorbani, Ali A. "Intrusion detection evaluation dataset (CIC-IDS2017)". *Proceedings of the Canadian Institute for Cybersecurity* (2018).

19. Zhang Tinget al. "A fusing framework of shortcut convolutional neural networks". *Information Sciences* 579 (2021): 685–699. https://doi.org/10.1016/j.ins.2021.08.030

20. Rehman Sadaqat Uret al. "Optimization of CNN through novel training strategy for visual classification problems". *Entropy* 20.4 (2018): 290. https://doi.org/10.3390/e20040290 PMID: 33265381

21. ur Rehman Sadaqatet al. "Unsupervised pre-trained filter learning approach for efficient convolution neural network". *Neurocomputing* 365 (2019): 171–190. https://doi.org/10.1016/j.neucom.2019.06.084

22. Liu X., Wang S., Lu S., Yin Z., Li X., Yin L., Zheng W. (2023). Adapting Feature Selection Algorithms for the Classification of Chinese Texts. *Systems*, 11(9), 483. https://doi.org/10.3390/systems11090483

23. Tayir T., & Li L. (2024). Unsupervised Multimodal Machine Translation for Low-resource Distant Language Pairs. *ACM Transactions on Asian Low-Resource Language Information Processing*, 23(4), 1–22. https://doi.org/10.1145/3652161

24. Hossain Md Alamgir, and Islam Md Saiful. "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning". *Array* 19 (2023): 100306. https://doi.org/10.1016/j.array.2023.100306

25. Yang Yanqinget al. "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization". *IEEE Access* 8 (2020): 42169–42184. https://doi.org/10.1109/ACCESS.2020.2977007

26. Di Yiet al. "A maneuvering target tracking based on fastIMM-extended Viterbi algorithm". *Neural Computing and Applications* (2023): 1–10.

27. Andresini Giuseppinaet al. "Multi-Channel Deep Feature Learning for Intrusion Detection". *IEEE Access* (2020). https://doi.org/10.1109/ACCESS.2020.2980937

28. Liu H., Jiang K., Gamboa H., Xue T., & Schultz T. (2022). Bell Shape Embodying Zhongyong: The Pitch Histogram of Traditional Chinese Anhemitonic Pentatonic Folk Songs. *Applied Sciences*, 12(16), 8343. https://doi.org/10.3390/app12168343

29. Jin C., Monfort A., Chen F., Xia N., & Wu B. (2024). Institutional Investor ESG Activism and Corporate Green Innovation Against Climate Change: Exploring Differences Between Digital and Non-Digital Firms. *Technological Forecasting and Social Change*, 200, 123129. https://doi.org/10.1016/j.techfore.2023.123129

30. He C., Huang K., Lin J., Wang T., & Zhang Z. (2023). Explain systemic risk of commodity futures market by dynamic network. *International Review of Financial Analysis*, 88, 102658. https://doi.org/10.1016/j.irfa.2023.102658

31. Li X., & Sun Y. (2021). Application of RBF neural network optimal segmentation algorithm in credit rating. *Neural Computing and Applications*, 33(14), 8227–8235. https://doi.org/10.1007/s00521-020-04958-9

32. Guo R., Liu H., & Liu D. (2024). "When deep learning-based soft sensors encounter reliability challenges: a practical knowledge-guided adversarial attack and its defense". *IEEE Transactions on Industrial Informatics*, 20(2), 2702–2714. https://doi.org/10.1109/TII.2023.3297663

33. Cai L., Yan S., Ouyang C., Zhang T., Zhu J., Chen L., et al. (2023). Muscle synergies in joystick manipulation. *Frontiers in Physiology*, 14. https://doi.org/10.3389/fphys.2023.1282295 PMID: 37900948

34. Zhang R., Li L., Zhang Q., Zhang J., Xu L., Zhang B., et al. (2023). Differential Feature Awareness Network within Antagonistic Learning for Infrared-Visible Object Detection. *IEEE Transactions on Circuits and Systems for Video Technology*. https://doi.org/10.1109/TCSVT.2023.3289142

35. Di Y., Li R., Tian H., Guo J., Shi B., Wang Z., et al. (2023). A maneuvering target tracking based on fastIMM-extended Viterbi algorithm. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-023-09039-1

36. Li J., Li J., Wang C., Verbeek F. J., Schultz T., Liu H. (2023). Outlier detection using iterative adaptive mini-minimum spanning tree generation with applications on medical data. *Frontiers in Physiology*, 14. https://doi.org/10.3389/fphys.2023.1233341 PMID: 37900945

37. Jiang Huiet al. "Network intrusion detection based on PSO-XGBoost model". *IEEE Access* 8 (2020): 58392–58401. https://doi.org/10.1109/ACCESS.2020.2982418

38. Hidayat Imran, Ali Muhammad Zulfiqar, and Arshad Arshad. "Machine Learning-Based Intrusion Detection System: An Experimental Comparison". *Journal of Computational and Cognitive Engineering* 2.2 (2023): 88–97. https://doi.org/10.47852/bonviewJCCE2202270

39. Yang H., Zhang X., Li Z., & Cui J. (2022). Region-Level Traffic Prediction Based on Temporal Multi-Spatial Dependence Graph Convolutional Network from GPS Data. *Remote Sensing*, 14(2), 303. https://doi.org/10.3390/rs14020303

40. Yu J., Lu L., Chen Y., Zhu Y., & Kong L. (2021). An Indirect Eavesdropping Attack of Keystrokes on Touch Screen through Acoustic Sensing. *IEEE Transactions on Mobile Computing*, 20(2), 337–351. https://doi.org/10.1109/TMC.2019.2947468

41. Liu D., Cao Z., Jiang H., Zhou S., Xiao Z., Zeng F. (2022). Concurrent Low-Power Listening: A New Design Paradigm for Duty-Cycling Communication. *ACM Transactions on Sensor Networks*, 19(1). https://doi.org/10.1145/3517013

42. Jiang H., Wang M., Zhao P., Xiao Z., & Dustdar S. (2021). A Utility-Aware General Framework With Quantifiable Privacy Preservation for Destination Prediction in LBSs. *IEEE/ACM Transactions on Networking*, 29(5), 2228–2241. https://doi.org/10.1109/TNET.2021.3084251

43. Chen Y., Zhu L., Hu Z., Chen S., & Zheng X. (2022). Risk Propagation in Multilayer Heterogeneous Network of Coupled System of Large Engineering Project. *Journal of Management in Engineering*, 38(3), 4022003. https://doi.org/10.1061/(ASCE)ME.1943-5479.0001022

44. Alosaimi Shema, and Almutairi Saad M. "An Intrusion Detection System Using BoT-IoT". *Applied Sciences* 13.9 (2023): 5427. https://doi.org/10.3390/app13095427

45. Li S., Chen H., Chen Y., Xiong Y., & Song Z. (2023). Hybrid Method with Parallel-Factor Theory, a Support Vector Machine, and Particle Filter Optimization for Intelligent Machinery Failure Identification. *Machines*, 11(8), 837. https://doi.org/10.3390/machines11080837

46. Zheng W., Deng P., Gui K., & Wu X. (2023). An Abstract Syntax Tree based static fuzzing mutation for vulnerability evolution analysis. *Information and Software Technology*, 107194. https://doi.org/10.1016/j.infsof.2023.107194

47. Tu Shanshanet al. "ModPSO-CNN: an evolutionary convolution neural network with application to visual recognition". *Soft Computing* 25 (2021): 2165–2176. https://doi.org/10.1007/s00500-020-05288-7

48. Tu Shanshanet al. "Optimisation-based training of evolutionary convolution neural network for visual classification applications". *IET Computer Vision* 14.5 (2020): 259–267. https://doi.org/10.1049/iet-cvi.2019.0506

49. Latif Jahanzaibet al. "ODGNet: a deep learning model for automated optic disc localization and glaucoma classification using fundus images". *SN Applied Sciences* 4.4 (2022): 98. https://doi.org/10.1007/s42452-022-04984-3

50. Latif Jahanzaibet al. "Digital forensics use case for glaucoma detection using transfer learning based on deep convolutional neural networks". *Security and Communication Networks* 2021 (2021): 1–13. https://doi.org/10.1155/2021/4494447

51. Rehman Obaid Uet al. "Design optimization of electromagnetic devices using an improved quantum inspired particle swarm optimizer". *The Applied Computational Electromagnetics Society Journal (ACES)* (2018): 951–956.

52. Rehman, Sadaqat ur et al. "Deep learning models for intelligent healthcare: implementation and challenges". In *Artificial Intelligence and Security: 7th International Conference*, *ICAIS 2021*, *Dublin*, *Ireland*, *July 19–23*, *2021*, *Proceedings*, *Part I*, pp. 214-225. Springer, 2021.

53. ur Rehman, Sadaqat et al. "Learning a semantic space for modeling images, tags and feelings in cross-media search". In *Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2019 Workshops*, *BDM*, *DLKT*, *LDRC*, *PAISI*, *WeL*, *Macau*, *China*, *April 14–17*, *2019*, *Revised Selected Papers*, pp. 65-76. Springer, 2019.

54. Akhtar Muhammad Shoaib, and Feng Tao. "Malware Analysis and Detection Using Machine Learning Algorithms". *Symmetry* 14.11 (2022): 2304. https://doi.org/10.3390/sym14112304

55. Alalayah Khaled M. et al. "Design an Internet of Things Standard Machine Learning Based Intrusion Detection for Wireless Sensing Networks". *Journal of Nanoelectronics and Optoelectronics* 18.2 (2023): 217–226.

56. Su Tongtonget al. "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset". *IEEE Access* 8 (2020): 29575–29585. https://doi.org/10.1109/ACCESS.2020.2972627

57. Ma J., & Hu J. (2022). Safe consensus control of cooperative-competitive multi-agent systems via differential privacy. *Kybernetika*, 58(3), 426–439. https://doi.org/10.14736/kyb-2022-3-0426

58. Chen B., Hu J., Zhao Y., & Ghosh B. K. (2022). Finite-Time Velocity-Free Rendezvous Control of Multiple AUV Systems With Intermittent Communication. *IEEE Transactions on Systems*, *Man*, *and Cybernetics: Systems*, 52(10), 6618–6629. https://doi.org/10.1109/TSMC.2022.3148295

59. Jiang Kaiyuanet al. "Network intrusion detection combined hybrid sampling with deep hierarchical network". *IEEE Access* 8 (2020): 32464–32476. https://doi.org/10.1109/ACCESS.2020.2973730

60. Nagaraja Arunet al. "Similarity Based Feature Transformation for Network Anomaly Detection". *IEEE Access* (2020). https://doi.org/10.1109/ACCESS.2020.2975716

61. Karatas Gozde, Demir Onder, and Sahingoz Ozgur Koray. "Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset". *IEEE Access* 8 (2020): 32150–32162. https://doi.org/10.1109/ACCESS.2020.2973219

62. Zolanvari Maedeet al. "Machine learning-based network vulnerability analysis of industrial Internet of Things". *IEEE Internet of Things Journal* 6.4 (2019): 6822–6834. https://doi.org/10.1109/JIOT.2019.2912022

63. Yu Yingwei, and Bian Naizheng. "An intrusion detection method using few-shot learning". *IEEE Access* 8 (2020): 49730–49740. https://doi.org/10.1109/ACCESS.2020.2980136

64. Xiao Yihanet al. "An intrusion detection model based on feature reduction and convolutional neural networks". *IEEE Access* 7 (2019): 42210–42219. https://doi.org/10.1109/ACCESS.2019.2904620

65. Guo C., & Hu J. (2023). Time base generator based practical predefined-time stabilization of high-order systems with unknown disturbance. *IEEE Transactions on Circuits and Systems II: Express Briefs*. https://doi.org/10.1109/TCSII.2023.3242856

66. Kasongo Sydney Mambwe, and Sun Yanxia. "A deep learning method with filter based feature engineering for wireless intrusion detection system". *IEEE Access* 7 (2019): 38597–38607. https://doi.org/10.1109/ACCESS.2019.2905633

67. Lee Jonghoonet al. "Cyber threat detection based on artificial neural networks using event profiles". *IEEE Access* 7 (2019): 165607–165626. https://doi.org/10.1109/ACCESS.2019.2953095

68. Bagaa Miloudet al. "A machine learning security framework for IoT systems". *IEEE Access* 8 (2020): 114066–114077. https://doi.org/10.1109/ACCESS.2020.2996214

69. Thakkar Ankit, and Lohiya Ritika. "Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System". *Information Fusion* 90 (2023): 353–363. https://doi.org/10.1016/j.inffus.2022.09.026

70. Jiang Y., & Li X. (2022). Broadband cancellation method in an adaptive co-site interference cancellation system. *International Journal of Electronics*, 109(5), 854–874. https://doi.org/10.1080/00207217.2021.1941295

71. Zheng C., An Y., Wang Z., Qin X., Eynard B., Bricogne M., et al. (2023). Knowledge-based engineering approach for defining robotic manufacturing system architectures. *International Journal of Production Research*, 61(5), 1436–1454. https://doi.org/10.1080/00207543.2022.2037025

72. Li H., Huang Q., Huang J., & Susilo W. (2023). Public-Key Authenticated Encryption With Keyword Search Supporting Constant Trapdoor Generation and Fast Search. *IEEE Transactions on Information Forensics and Security*, 18, 396–410. https://doi.org/10.1109/TIFS.2022.3224308

73. Lyu T., Xu H., Zhang L., & Han Z. (2024). Source Selection and Resource Allocation in Wireless-Powered Relay Networks: An Adaptive Dynamic Programming-Based Approach. *IEEE Internet of Things Journal*, 11(5), 8973–8988. https://doi.org/10.1109/JIOT.2023.3321673

74. Gao N., Han Y., Li N., Jin S., & Matthaiou M. (2024). When Physical Layer Key Generation Meets RIS: Opportunities, Challenges, and Road Ahead. *IEEE Wireless Communications*. https://doi.org/10.1109/MWC.013.2200538

75. Zhao D., Cai W., & Cui L. (2024). Adaptive thresholding and coordinate attention-based tree-inspired network for aero-engine bearing health monitoring under strong noise. *Advanced Engineering Informatics*, 61, 102559. https://doi.org/10.1016/j.aei.2024.102559

76. Martins Nunoet al. "Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review". *IEEE Access* (2020). https://doi.org/10.1109/ACCESS.2020.2974752

77. Zheng W., Lu S., Cai Z., Wang R., Wang L., Yin L. (2024). PAL-BERT: An Improved Question Answering Model. *Computer Modeling in Engineering & Sciences*, 139(3), 2729–2745. https://doi.org/10.32604/cmes.2023.046692

78. Li M., Li L., Tao X., Xie Z., Xie Q., Yuan J. (2024). Boosting Healthiness Exposure in Category-constrained Meal Recommendation Using Nutritional Standards. *ACM Transactions on Intelligent Systems and Technology*. https://doi.org/10.1145/3643859

79. Thakkar Ankit, and Lohiya Ritika. "A review of the advancement in intrusion detection datasets". *Procedia Computer Science* 167 (2020): 636–645. https://doi.org/10.1016/j.procs.2020.03.330

80. Sangaiah Arun Kumaret al. "A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in cloud of things". *Cluster Computing* 26.1 (2023): 599–612. https://doi.org/10.1007/s10586-022-03629-9