electronics

Article

A Meta-Model to Predict and Detect Malicious Activities in 6G-Structured Wireless Communication Networks

Haider W. Oleiwi [1,*], Doaa N. Mhawi [2] and Hamed Al-Raweshidy [1]

[1] Department of Electronic and Electrical Engineering, Brunel University London, London UB8 3PH, UK
[2] Technical Institute for Administration, Middle Technical University, Baghdad 10010, Iraq
* Correspondence: haider.al-lami@brunel.ac.uk

Abstract: The rapid leap in wireless communication systems incorporated a plethora of new features and challenges that accompany the era of 6G and beyond being investigated and developed. Recently, machine learning techniques were widely deployed in many fields, especially wireless communications. It was used to improve network traffic performance regarding resource management, frequency spectrum optimization, latency, and security. The studies of modern wireless communications and anticipated features of ultra-densified ubiquitous wireless networks exposed a risky vulnerability and showed a necessity for developing a trustworthy intrusion detection system (IDS) with certain efficiency/standards that have not yet been achieved by current systems. IDSs lack acceptable immunity against repetitive, updatable, and intelligent attacks on wireless communication networks, significantly concerning the modern infrastructure of 6G communications, resulting in low accuracies/detection rates and high false-alarm/false-negative rates. For this objective principle, IDS system complexity was reduced by applying a unique meta-machine learning model for anomaly detection networks was developed in this paper. The five main stages of the proposed meta-model are as follows: the accumulated datasets (NSL KDD, UNSW NB15, CIC IDS17, and SCE CIC IDS18) comprise the initial stage. The second stage is preprocessing and feature selection, where preprocessing involves replacing missing values and eliminating duplicate values, leading to dimensionality minimization. The best-affected subset feature from datasets is selected using feature selection (i.e., Chi-Square). The third step is represented by the meta-model. In the training dataset, many classifiers are utilized (i.e., random forest, AdaBoosting, GradientBoost, XGBoost, CATBoost, and LightGBM). All the classifiers undergo the meta-model classifier (i.e., decision tree as the voting technique classifier) to select the best-predicted result. Finally, the classification and evaluation stage involves the experimental results of testing the meta-model on different datasets using binary-class and multi-class forms for classification. The results proved the proposed work's high efficiency and outperformance compared to existing IDSs.

Keywords: 6G wireless communications; chi-square; cybersecurity; intrusion detection system; machine learning techniques; meta-model; stacking ensemble learning; voting techniques

Citation: Oleiwi, H.W.; Mhawi, D.N.; Al-Raweshidy, H. A Meta-Model to Predict and Detect Malicious Activities in 6G-Structured Wireless Communication Networks. *Electronics* **2023**, *12*, 643. https://doi.org/10.3390/electronics12030643

Academic Editors: Shihao Yan, Guanglin Zhang, Li Sun, Tsz Hon Yuen, YoHan Park, Changhoon Lee and Tao Huang

Received: 3 January 2023
Revised: 20 January 2023
Accepted: 24 January 2023
Published: 28 January 2023

1. Introduction

The advancement of modernized wireless communication networks with their accompanying features, technologies, heterogeneously connected networks/gadgets, service demands, and the huge amount of data traffic has brought more complexity and sophistication to communication systems [1]. The 6G revolution and internet of everything (IoE) technology drive artificial intelligence (AI)-based incorporations (e.g., machine learning (ML)) in the ubiquitous connection of billions of sub-networks, users, and devices. Furthermore, the new features of 6G and beyond wireless communications, movable infrastructure, and the potential intelligent services add critical security risks to the network's core, edge, and associated devices [1–4]. Modern networks benefit significantly from AI and ML in various ways, such as intelligent communications, network optimization, and big data analytics. However, the threats of renewable intelligent attacks on the networks increase proportionally with the complexity increase (caused by heterogeneity, enormous scale, and variety of applications these networks serve) [5–10]. The difficulty of creating adequate security procedures to defend the network increases due to the possibility of attackers discovering network vulnerabilities utilizing AI techniques. Thus, it is highly necessary to build a robust intelligent intrusion detection system (IDS) to comply with the evolution of intelligent attacks and to secure future networks [11–15]. The new networks connect a variety of billions of users/devices to serve people, providing a plethora of services/applications via the network's main components, e.g., the base station (BS) using the edge of technologies, e.g., terahertz communications, non-orthogonal multiple access, and IoE [12,15,16]. In risk-sensitive systems safety, the realization of a zero-day attack is not an easy process, especially with the proliferation of numerous malicious activities. Figure 1 demonstrates a sample of the 6G general expected infrastructure with a number of nominated applications and media over different areas [17].

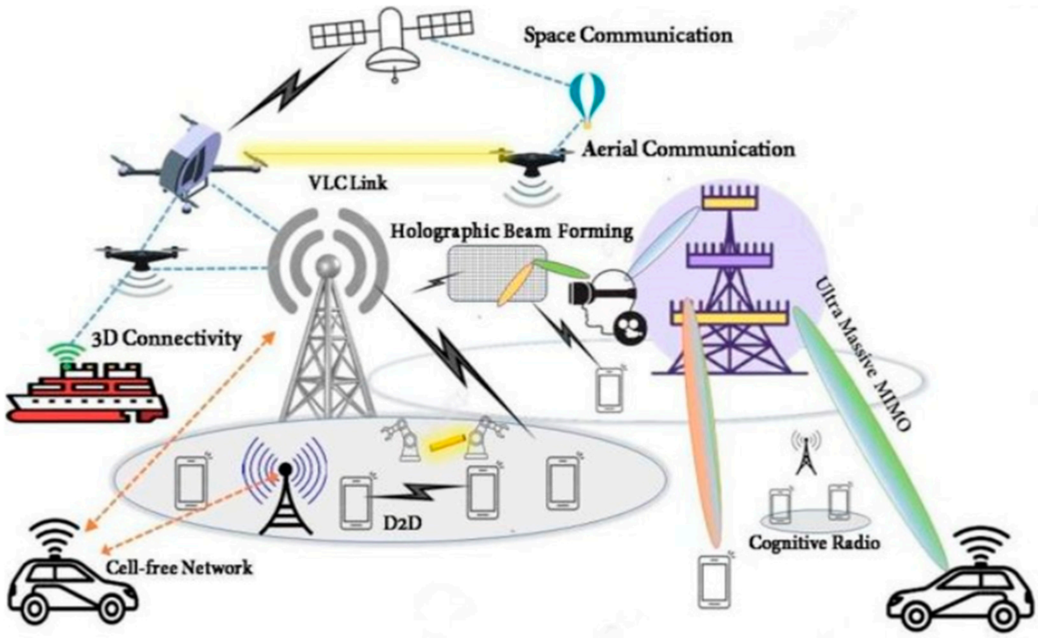

Figure 1. A sample of 6G expected infrastructure and applications.

IDSs send out notifications when discovering an unexpected activity or identified hazards. Any destructive behavior that interferes with the information system is considered an intrusion [18]. IDSs scan computers for unusual activities a conventional packet filter may fail. IDSs note any indicator for potentially dangerous action of network packets, as well as signals for highly resilient cyber defenses against disruptive activities and non-authorized access to a computer system. IDSs use two methods to detect intrusions (i.e., misuse and anomaly). A new IDS that includes these two methods was presented to overcome these limitations to increase accuracy and decrease FAR [11,19–25]. Furthermore; feature selection (FS) is a useful approach for IDSs to specify the significant features and cancel the useless features with less performance degradation [26–28]. IDSs require classifier methods to detect the final results and there are different AI methods for this task, e.g., ensemble learning (EL). EL techniques were used as building blocks for more complicated models by integrating many weak learners in EL methods, e.g., Bagging, boosting, AdaBoosting, and stacking (meta-model). These models of classifiers are used to reduce variance when using the bagging method, manipulated high bias to achieve strong classifiers inside these models when using the boosting, and the main session of the stacking (meta-model) is to combine the strengths of several effective models to provide predictions that perform better than any one model in EL [29].

However; IDSs still do not achieve the needed optimization for detection rate (DR), false alarm rate (FAR), or running time because of the high-dimensional dataset and abundant Zero-day attacks. Despite having a direct influence on resources, time complexity was not given as a significant consideration. Besides, the technological realm is envisioning IoE and 6G networks depending on the equipment that is programmed using lightweight algorithms.

This work targets initiating more sufficient/robust ML techniques-based attack-resistant detection to increase the IDSs' stability and accuracy by reducing the amount of computation/time needed by using four different datasets. The proposed model trains the FS method and ML algorithms to realize accurate/efficient IDs. Utilizing AI systems, the orientation of wireless communications must be thought about. Therefore; the contributions of this work are:

- In the context of FS and preprocessing, we used the Chi-square method for cleaning and preparing four different unbalanced datasets (NSL_KDD, UNSW _NB15, CIC_IDS17, and SCE_CIC_IDS18) to select the best subset features. Furthermore; enhancing the effectiveness of the training and testing stages is much more advantageous. These datasets undergo the cleansing and selection processes to select only the affected features to reduce time and achieve the best accuracy result.
- We enhance the performed effectiveness of the multiclass and binary class forms used with the four imbalanced datasets. Hence, the proposed work presents a novel meta-model that uses different ML techniques (i.e., random forest RF, Gradient Boost, AdaBoosting, LightGBM, XGBoosting, and CatBoosting) to work as a base classifier and then applies the meta-model technique using decision tree (DT) to select the best-affected result (prediction). The meta-model works as a prediction method to select only the classifiers with high accuracy and then enter the results into the testing part to achieve the final result.

The remaining sections of this paper are organized as follows:

Section 2 implies several similar works, while Section 3 provides a detailed definition of the proposed system's methodology and addresses the experimental findings. Furthermore, it illustrates how the proposed method was implemented with the applied datasets and addresses the technical constraints. Finally, the conclusions are stated in Section 4, which summarizes the results, directions for further investigation, and future suggestions.

2. Literature Review

In this section, the authors study the other related similar studies and demonstrate them in Table 1 for better understandable readability. Furthermore, to distinguish each of those related studies the main FS method with the number of FSs, type of the classification method, experimental results, and disadvantages.

Table 1. Similar related studies.

References/Authors	FS Methods and Number of Features	Classifiers Methods	Experimental Results	Cons
[11], Oleiwi et. al.	They used correlation FS combined with RF EL. This system selected 30, 35, and 40 FSs for (NSL, UNSW_NB, AND CIC_IDS) respectively.	Adopting two modified classifiers (RF and SVM) and applying the classifiers as AdaBoosting and bagging EL; then aggregating these classifiers by the voting average technique.	The experimental results are 99.6% accuracy with 0.004 FAR for NSL_KDD, 99.1% accuracy with 0.008 FAR for UNSW_NB2015, and 99.4% accuracy with 0.0012 FAR for CIC_IDS2017.	Complexity time measurement took too much time, due to the merging of two methods of EL techniques for splitting and disseminating normal or suspicious network traffic attacks.
[30], Gaikwad, D. and Thool, R.	N/A	DT and rule learner-based EL.	It shows that the classifiers methods of IDS exhibit the lowest false positive rate (FPR) with higher classification accuracy (i.e., 80%, 81%, 15.1%) for (accuracy, DR, FAR).	Not accurate results and undetected several attacks. Furthermore; A long time for searching with the lowest accuracy and false negative rate (FNR).
[31], Pajouh et. al.	linear discriminant analysis (They have chosen 16 features).	Two-tier anomaly-detection model using K-Nearest Neighbor KNN.	The experimental evaluation of 83.24% accuracy, 4.83% FAR, 82% true positive rate (TPR), and 5.43 FPR.	needed more execution time. Insufficient dealing with the network imbalance of anomaly datasets.

Table 1. *Cont.*

References/Authors	FS Methods and Number of Features	Classifiers Methods	Experimental Results	Cons
[32], Kanakarajan, N.K. and Muniasamy, K.	Information gain adopts 32 features for binary class and with 10-features for multiclass.	Hybrid RF with Adaptive Greedy randomized.	Accuracy is 85.0559% with information gain reaching an accuracy of 78.9035%.	Less accuracy and high FAR.
[33], Mittal, M. et. al.	DT for FS.	ML techniques for energy efficiency and anomaly detection in hybrid wireless sensor networks.	The experimental results showed that accuracy is 95%, where the precision is 94.00%, recall is 98.00%, and F1-Score is 96.00%.	Long time for searching and A high FAR.
[34], Jaw, E. and Wang, X.	The wrapper method is based on a genetic algorithm to select (11, 8, and 13) features.	Different classifiers are used for classification.	The results showed 98.99% for CIC_IDS17, 98.73% for NSL_KDD, 97.997% for UNSW_NB15 accuracy, with 98.75%, 96.64%, 98.93% DRs.	Not accurate results and undetected several attacks. A long time for searching. Furthermore; low FNR.
[35], Gupta, N. et. al.	RF was adopted to select the best subset features. By used NSL_KDD, CIDDS-001, and CIC_IDS2017.	The extreme gradient Boosting algorithm is used as a classifier with deep learning.	The experimental results are 99% for NSL, 96% for CICIDS-001%, and 92% for CIC_IDS2017.	Complexity time measurement has taken several hours, due to the deep learning techniques for splitting and disseminating normal or suspicious network traffic attacks.
[23], Mhawi. et. al.	Hybrid of Correlation FS coupled with Forest Panelized Attributes.	They used four different classifiers (i.e., SVM, RF, Naïve Bayes NB, and K-Nearest-Neighbor).	The experimental results are 99.7% for CIC_IDS17 of accuracy with 0.0053 FNR, and 0.004 FAR.	Complexity system in the FS stage and classification stage. It takes high time in the training part.

To the researchers' knowledge, the provided system outperforms the earlier systems in terms of performance and outcomes. Using numerous datasets, it considerably excels in literature performance and delivers the highest results.

3. Methodology

IDSs observe malicious or suspicious activities in the traffic across the whole communication network. They were presented to wireless communication networks to examine for any abnormal activity occurring throughout control/data communication. The hacker attempts to penetrate networks to stop communications or capture important data. By breaching networks' security and affecting the behaviors of sensors/networks, the attacker inserts bugs into a network. To solve this sensitive issue and protect the system from malicious actors, a properly secured framework is required. The proposal's main structure is shown in Figure 2.

Figure 2 shows different stages to detect suspicious/malicious activities (anomalies) over the communication network undergoing preprocessing. Before these stages, collecting different types of datasets and detecting the missing values are required, replacing the null values with some values, while average values are considered. After that, duplicate values are deleted from datasets (NSL_KDD, UNSW_NB15, CICI_IDS17, and SCE_CIC_IDS18).

Next step, data normalization and encoding processes are performed. Encoded data undergoes a dimensionality decrease to aid data handling. Accordingly, features are optimized to attain the optimal features out of the entire data. This is helpful to detect anomalies within data. After preprocessing, the cleansed data will transfer to the next level to utilize impacted features only to the finalized results by applying Chi-square. Ultimately, the proposed system uses meta-ML models as a classifier to detect and predict malicious activities in the network traffic. It includes a number of stages that include several steps with a dedicated task each. Each stage's outcome represents an input to its next stage. The stages are described in detail successively.

3.1. First Stage: Datasets Collection

The researchers' main problem is finding an appropriate dataset for evaluating IDSs. Therefore; there are different collected datasets used with different features (NSL_KDD, UNSW_NB15, CIC_IDS17, and SCE_CIC_IDS18). They were collected from different sites and contained different types of attacks. These datasets are used for experiments, and each dataset is briefly described as follows:

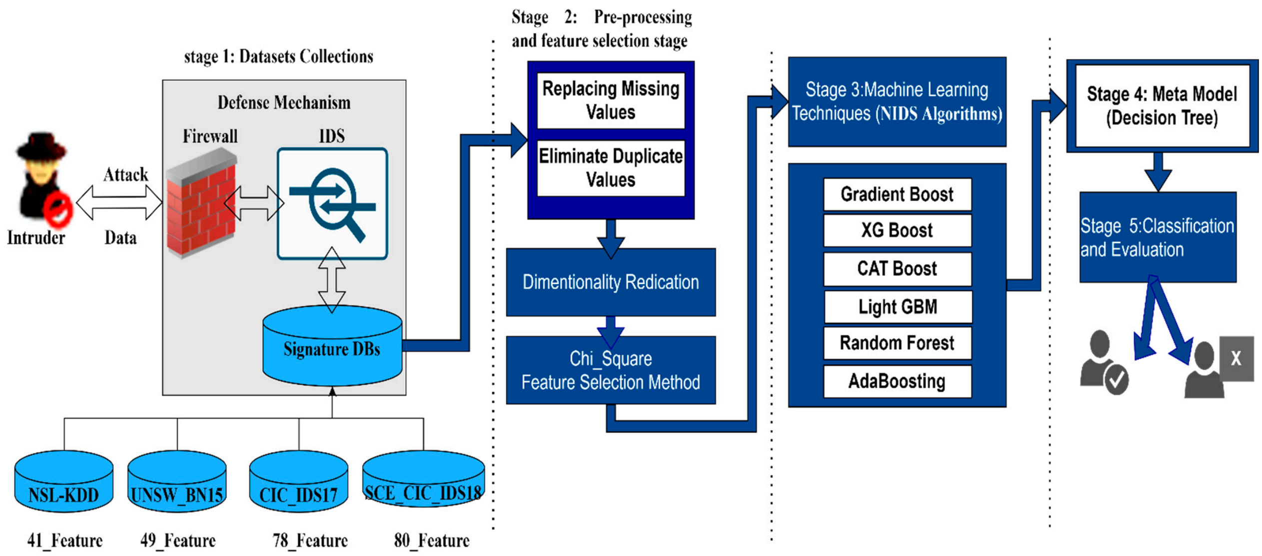

Figure 2. The proposed system's general structure.

3.1.1. First: NSL_KDD Dataset

NSL-KDD is a dataset suggested to solve some of the inherent problems of the KDD'99 dataset. Because of the scarcity of freely available datasets for networking-built IDSs, the new dataset's version is still in service as a high-impact benchmark dataset to help the researchers in comparison of multiple ID strategies, although they have technical issues noted by McHugh. NSL-KDD training set and testing set have a notable quantity of records. The achieved gain enables cost-effective experimentation on the entire set without arbitrary selection of a limited subset.

3.1.2. Second: UNSW_NB15 Dataset

It is a network intrusion dataset that is collected by the university of the new southern western network base in 2015. It contains nine types of attacks. Raw network packets are included in the dataset. There are 175,341 records in the train set and 82,332 records from various types of activities in the test set (attacks and normal activities).

3.1.3. Third: CIC_IDS17 Dataset

The CIC_IDS17 dataset (compiled in 2017) was released by the Canadian Institute for Cybersecurity (CIC). It offers positive information and the most current widespread attacks. The outcomes of the network traffic analysis using the CIC flow meter are also presented. Time-stamped flows exist for protocols, source/destination IPs, ports, and attacks. One of the most recent datasets is this one. Updated DDoS, Brute Force, XSS, SQL Injection, Infiltration, Port Scan, and Botnet assaults are among the things it contains. There are 2,830,743 records total in this dataset, which is divided into eight files. Each record comes with 78 unique characteristics and labels. In order to maintain the same magnitude order for each dataset when multi-classification is required.

3.1.4. Fourth: SCE_CIC_IDS18 Dataset

The University of New Brunswick created this dataset for analyzing DDoS data. It was sourced completely from 2018 and stopped updates. The dataset was built depending on the university's servers' logs, which have observed a variety of DoS attacks during the free availability era. When writing the dataset, ML notebooks observed that the label column is the precious portion, as it determines if the transmitted packets are malicious or benign. Data is divided into various files based on date. Each file is unbalanced, and it is up to the notebook creator to divide the dataset into a balanced form for higher-quality predictions. It has eighty columns, each of which corresponds to an entry in the IDS logging system the University of New Brunswick has. Given the system divides traffic into forward and backward. The most important columns within this dataset (i.e., Destination port, Protocol, Flow Duration, total forward packets (Tot Fwd Pkts), total backward packets (Tot Bwd Pkts), and label (Label).

3.2. Second stage: Preprocessing and FS

The datasets collected in the first stage undergo preprocessing and FS steps. The processing of these steps is demonstrated in Algorithm 1.

Algorithm 1. Preprocessing and FS.

Input: Reading Four different Datasets [] = [D1, D2, D3, and D4], N = sample size.
Output: $BestFeature$.
Begin
LOOP:
Repeat from 1 to N
1. Preprocessing steps:
(Filteration process):
Reading Datasets [i]
Repeat
  If Datasets [i] = np. information or -np. information then
  Datasets [i] = NAN */np,-np are negative, positive infinity */
  If Datasets [i] = Missing_values and duplicated_values then
  Datasets [i] = dropping values.
(Transformation process):
  If the Datasets [i] = nonnumerical_values then
  Call One_Hot_encoding function then return new datasets [i].
Normalization (Computing MinMax Scal function):
  Check Call Min_value [i] function for each dataset [i].
  Check Call Max_value [i] function for each dataset [i].
  $\text{XiValue}[\text{i}] = \frac{\text{XiValue}-\text{Min\_value}[\text{i}]}{\text{Max}_{\text{value}}[\text{i}]-\text{Min}}$
Until Datasets [i] greater than N;
Return XiValue[i].
End Loop
2. Feature_Selection steps:
For each dataset [i] split XiValue[i] into two parts Training_part [i] and Testing_part [i]. */ 70%training_part and 30% testing_part */.
Repeat
DF = N − 1. (Freedom degrees (DF)) */It refers to the maximum number of logically independent values that can vary*/.
Compute each part Chi-square as follows: $x_2^c = \sum \frac{(Oi-Ci)^2}{Ei}$
End Loop
Return the best features Xi for each dataset [i].
End

In Algorithm 1, raw data in each dataset is passed into two main steps. Firstly, preprocessing to clean and prepared data (filtration process) then non-numerical values are converted into numerical using the one-hot encoding (transformation process) and then converted into the binary form using the Minimax scaling function (normalization). The outcome of this algorithm is to return the best subset features of each dataset. Therefore; the best subset features are (20, 30, 35, and 38) for NSL_KDD, UNSW_NB15, CIC_IDS17, and SCE_CIC_IDS18 datasets, respectively.

3.3. Third and Fourth Stages: ML Techniques for NIDS (Training Set) and Voting Techniques (Meta-Model) for the Testing Set

For the training stage, many different classifiers are used (i.e., XGBoosting, random forest (RF), AdaBoosting, GradientBoosting, LightGBM, and CatBoost) each of them considered as a base classifier. Each of these classifiers manipulates the training data independently by taking the Di of each dataset. Afterby, the results of each base classifier (predictions) are aggregated into the meta-model (DT), Figure 3 demonstrates the main idea of meta-model classifiers. Furthermore, the testing stage begins in the meta-model to get the prediction results to check the evaluation and performance of the proposed meta-model. Algorithm 2 illustrates this stage.

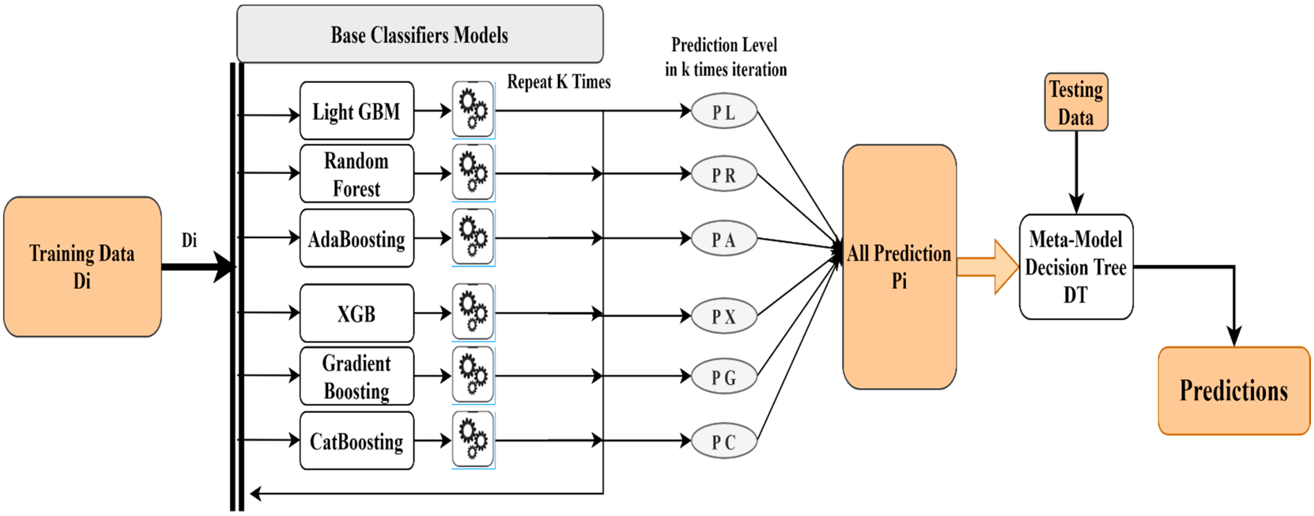

Figure 3. The Meta-model structure.

The meta-model working mechanism are demonstrated in detail in the following subsections.

3.3.1. The Datasets Partitioning Mechanism

It is necessary to aggregate the result of each classifier through the composite model and then send them to the stacking model to select the best result for voting. Furthermore, the voting technique is a type of EL methods that combines the predictions of several different models (classifiers) and selects the best prediction with the most votes.

As shown in Figure 3, the meta-model system has four traffic datasets, it uses three datasets as source datasets to train the meta-model, whereas the fourth dataset is used as a target to fine-tune it and then test the model performance. Each source dataset requires splitting into training and validation partitions. During training, it randomly selects two batches of samples from the training datasets, using one batch to compute the task-specific parameters and the other batch to compute the loss. Then repeat the same process with the validation dataset to be able to select the best prediction model. After the training, it is essential to fine-tune the model upon the target dataset.

3.3.2. Classifiers Work and Aggregation Techniques

In Algorithm 2 there are different classifiers, each of which performs a specific process and manipulates problems precisely. RF is a meta-estimator that fits several DT classifiers on different datasets' sub-samples, applying averaging to enhance predictive accuracy and controlling overfitting. Subsample and original input sample sizes are usually the same, however, samples are drawn with replacement if bootstrap = True. While XGBoost optimized gradient boosted DT. This classifier does not need normalized features and works well if the data is nonlinear, non-monotonic, or with segregated clusters. Whereas the AdaBoosting classifier is to fit a sequence of weak-learners (e.g., models that are better than stochastic guessing, like small DTs) on repetitively modifying data versions. Consequently, the predictions get integrated by a weighted majority vote (or sum) to generate the final prediction. Data modifications at each so-called boosting iteration include applying weights $\omega_1, \omega_2, \omega_3, \ldots \ldots, \omega_N$ to every training sample.

Algorithm 2. ML and meta-model techniques.

Input: Xi for each dataset [i] from Algorithm 1;
K /* is the number of classifiers*/;
Learning_Rate (LR);
Random_state (RS);
$\mathrm{M_i}$; /* Error rate of each classifier*/; (i.e., $(\mathrm{Mi}) = \sum_{j=1}^{d} wj \times err(Xj)$);
Number of Estimators (NS); /* subset number*/;
Criterion; /* type of measure*/;
Machin learning classifiers (Bse classifiers); (i.e.,
RandomForest (C1),
XGBoosting (C2),
AdaBoost (C3),
GradientBoosting (C4),
LightGBM (C5),
and CatBoosting (C6)).
Meta-model classifier (i.e., DT (C8))
Output: A composite model.
Begin
1. ML techniques (base-classifiers):
Read a number of K.
Loop: from 1 to k
RandomForest (C1) Determine attribute:
RS = 1, NS = 10, LR = 0.01, max_features [integer] /*The number of features to consider when looking for the best split*/.
XGBoosting classifiers (C2) Determine attribute:
Determine attribute: LR = 0.01, RS = 1.
AdaBoosting classifiers (C3) Determine attribute:
Determine attribute: RS = 1, and NS = 10, wi = 1/N.
GradientBoosting (C4) Determine attribute: (Loss = 'deviance', LR = 0.1, number of estimators = 100, minimum split samples = 2, maximum depth = 3, fraction of validation = 0.1).
LightGBM (C5) Determine attribute:
RS = 1, and NS = 10.
CatBoosting (C6) Determine attribute:
RS = 1, and NS = 10.
Repeat
For i = 1 to 6 do
Mi for the prediction by applying:
$\mathrm{M_i} = \sum_{j=1}^{d} wj \times err(Xj)$.
If $\mathrm{M_i}$ is larger than half then
$[\log(1 - (\mathrm{M_i}))/(\mathrm{M_i})]$.
End if
Until the results of 6 Ci
End for
Return all Ci with minimum $\mathrm{M_i}$.
2. Meta-model (DT) and compute (Voting techniques):
Repeat
Compute average weighting techniques for all Ci by $\frac{1}{\mathrm{mj}} = \mathrm{l}\sum_{\mathrm{i}=1}^{\mathrm{l}} \mathrm{pci}\left(\frac{\mathrm{wi}}{\mathrm{x}}\right)$.
Measurements of the binary and multi-class forms:
DR, FNR, FPR, TPR, TNR, accuracy, FAR, precision, and recall.
Until the result is the best.
Return composite-model.
End

Since all weights are initially set to ωi = 1/N, the initial step trains a learning algorithm using initial data. The sample weights are individually adjusted for each further iteration, and the learning process is then performed once more on the reweighted data. Furthermore; to compute and adjust weight, it undergoes the following steps:

- Assigning equal weights to all the data points to find the stump that does the best job, classifying the new collection of samples by finding their Gini Index and selecting the sample's weight with the lowest Gini index.
- Calculating the "Amount of Say" and "Total error" to update the previous sample weights.
- Normalizing the new sample weights.

The consequences of training examples at a particular stage are changed to reflect whether or not the boosted model that was induced in the preceding step accurately predicted those training examples. Examples that are challenging to foresee get growing importance during the iterative process. As a result, each weak learner after them in the chain is compelled to focus on the instances that they missed before. Using gradient-boosting tree strategies has numerous benefits, which include:

- Generally, more accurate compared to other classifiers models.
- Train faster, especially on larger datasets.
- Most of them provide support handling categorical features.
- Some of them handle missing values natively.
- Often provides unbeatable predictive accuracy.
- Plenty of flexibility could optimize various loss functions.
- Provides multiple hyper-parameter setting options, making the function fit very flexibly.

LightGBM is a fast-distributed high-performance gradient-boosting framework based on DT algorithms, it is used for ranking, classification, and many other ML tasks. The CatBoost classifier is an algorithm for gradient boosting on DTs. It is used for search, recommendation systems, personal assistants, self-driving cars, weather prediction, and many other tasks in different companies.

3.4. Fifth Stage: Implementation and Evaluation

3.4.1. Implementation

It is carried out by applying four datasets (NSL_KDD, UNSW_NB15, CIC_IDS17, and SCI_CIC_IDS18). The train portion is 70% while the test portion is 30% to evaluate the proposal.

System Performance is evaluated by implementing the proposal using four various features selected using chi-square. The intrusion is detected by using different ML techniques with multiclass and binary-class forms of confusion matrices. Ultimately, performance evaluation is done by using multiple measurements; recall, precision, DR, FAR, and FNR. It is carried out by anaconda python 3.9 software and colab platform with Sklearn, Kearse, and Tensor Flow libraries with laptop hardware with the: CPU Core i7, generation 10th, and 11 windows operating system with 64-bit.

3.4.2. Evaluation and Experimental Results

1 Binary-Class and Multi-Class Confusion-Matrix forms

The experiment is conducted at this stage of the ML and meta-model (voting techniques) using four different datasets. Confusion-matrix is adopted in each class, which includes benign and attack network traffic. Furthermore, four Features are applied to detect suspicious activities on the network traffic. The proposed system uses binary and multi-class forms confusion matrices.

The distribution of the four states of true-positive (TP), false-positive (FP), true-negative (TN), and false-negative (FN) with different numbers of FSs and computing accuracy and FNR are explained in Table 2.

Table 2 explains the best features and results of accuracy and FNR (i.e., false negative detections are classified into FN and TP detections in the experiment) when using NSL_KDD, UNSW_NB15, CIC_IDS17, and SCE_CIC_IDS18 are (20, 30, 35, and 38), respectively. This measurement is significant to measure the efficiency and professionalism of the proposal due to calculating the total number of errors found in every attack diagnosed as normal. additionally, applying other features leads to an insufficiency of FNR and accuracy measures.

Table 2. Accuracy and FNR for (NSL_KDD, UNSW_NB15, CIC_IDS17, and SCE_CIC_IDS18) datasets when applied to different FSs.

Datasets	FS	TP	TN	FP	FN	Accuracy	FNR
NSL_KDD	10	9000	2280	715	605	9000 + 2280/12,600 = 0.89	605/(605 + 9000) = 0.06
	20	9714	2885	1	0	9714 + 2885/12,600 = 0.99	0/(0 + 9714)= 0
	30	9500	2480	215	405	9500 + 2480/12,600 = 0.95	405/(405 + 9500) = 0.04
	all	1525	630	144	201	1525 + 630/2470 = 0.87	201/201 + 1525 = 0
UNSW_NB15	10	1500	400	226	344	1500 + 400/2470 = 0.76	344/344 + 1500 = 0.19
	20	1525	630	144	201	1525 + 630/2470 = 0.87	201/201 + 1525 = 0.11
	30	1701	744	0	25	1701 + 744/2470 = 0.99	25/25 + 1701 = 0
	all	1000	400	226	844	1000 + 400/2470 = 0.56	844/844 + 1000 = 0.45
CIC_IDS17	10	443,615	48,561	10,650	62,736	492,176/565,562 = 0.87	62,736/443,615 + 62,736 = 0.123
	20	437,550	86,556	16,715	24,741	524,106/565,562 = 0.92	24,741/24,741 + 437,550 = 0.053
	30	453,916	10,928	1349	369	453,916/565,562 = 0.99	369/1369 + 453,916 = 0.0008
	35	453,916	110,928	349	369	564,844/565,562 = 0.99	369/369 + 453,916 = 0
	40	453,890	111,048	249	357	564,938/565,562 = 0.98	249/454,247 = 0.0005
	50	437,550	86,556	16,715	24,741	524,106/565,562 = 0.92	24,741/24,741 + 437,550 = 0.053
	all	443,615	48,561	10,650	62,736	492,176/565,562 = 0.87	62,736/443,615 + 62,736 = 0.123
SCE_CIC_IDS18	10	100,000	142,945	42,439	27,971	242,945/313,426 = 0.77	27,971/(100,000 + 27,971) = 0.218
	20	127,945	142,439	42,000	971	270,384/313,426 = 0.86	971/137,655 = 0.0705
	30	127,945	152,539	32,000	871	280,484/313,426 = 0.89	871/871 + 127,945 = 0.006,76,158
	38	142,439	170,916	0	71	313,355/313,426 = 0.99	71/(71 + 142,439) = 0.000,02,1821
	40	127,945	152,539	32,000	871	280,484/313,426 = 0.89	871/871 + 127,945 = 0.006,76,158
	50	127,945	142,439	42,000	971	270,384/313,426 = 0.86	971/137,655 = 0.0705
	60	100,000	142,945	42,439	27,971	242,945/313,426 = 0.77	27,971/(100,000 + 27,971) = 0.218
	all	100,045	102,000	43,339	67,971	202,045/313,426 = 0.64	67,971/(100,045 + 67,971) = 0.40

The core objective of utilizing different datasets is to train the proposed system for different types of attacks and make it more robust against suspicious traffic activities. Figures 4 and 5 demonstrate the final results of the binary form and multiclass form of the confusion matrix.

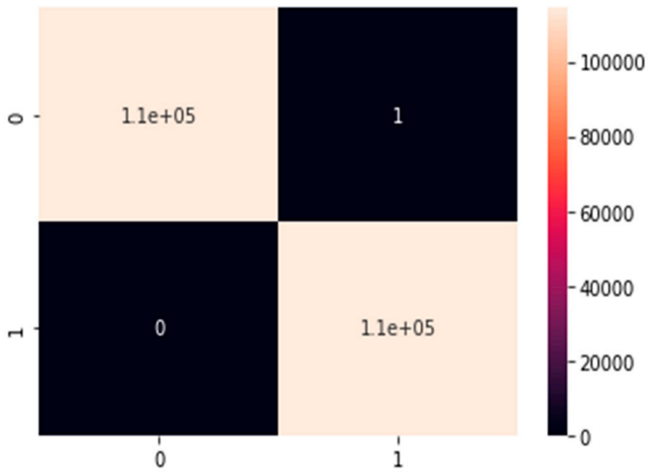

Figure 4. Binary-class confusion matrix.

Figure 4 shows that the proposed system achieves the best prediction results, it distinguishes benign activities and attacks precisely, and it can be noticed that only one percent of the benign activities is predicted as an attack; this result does not affect the final results.

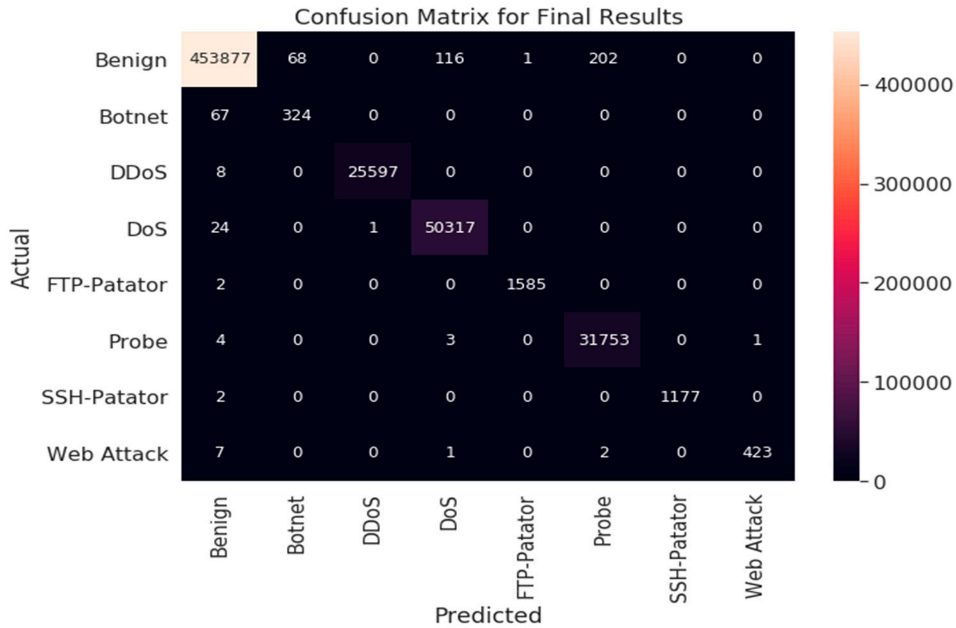

Figure 5. Multi-class confusion matrix.

In Figure 5, irrespective of the individual class's accuracy, the accuracy of the entire system (i.e., 99%) depends on the average accuracy of all the classes.

Furthermore; Figures 6 and 7 demonstrate the training and testing confusion matrix with the final measurements' results.

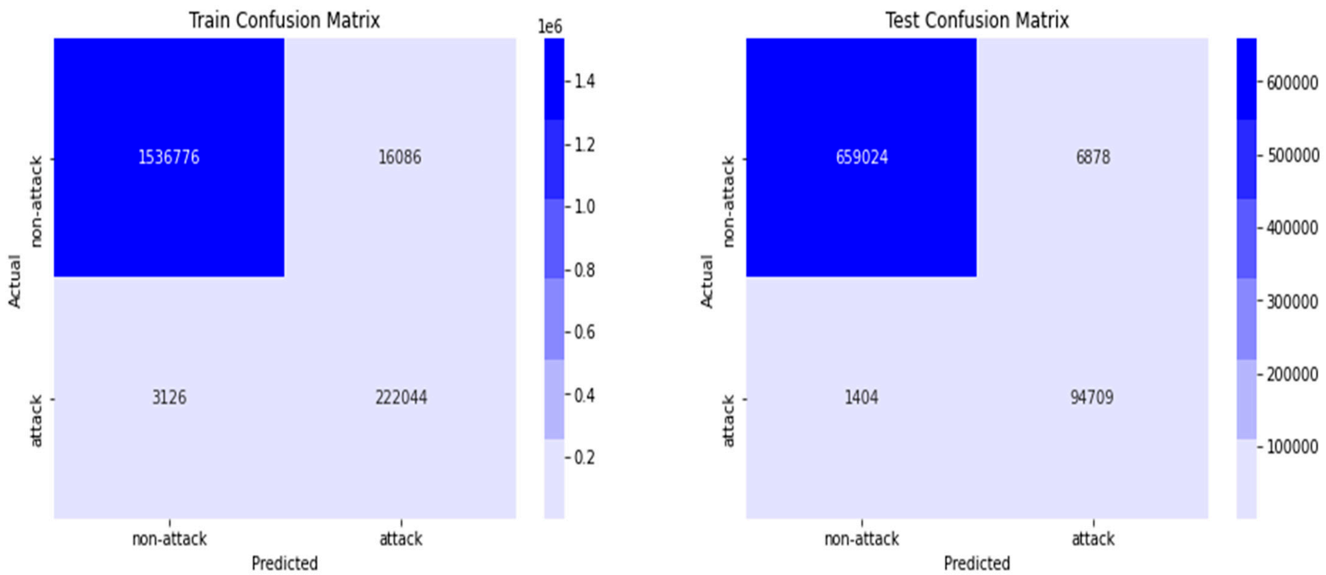

Figure 6. Train and Test confusion matrix.

2 BIG O Notation Measures

The complexity time of this proposed system is measured by applying the Big O notation (i.e., O (N^2)). It contains the calculations of complexity time. However, Figure 8 illustrates datasets classes with the required running time. Noticed the running time is increasing proportionally with input increase.

Figure 8 explains system complexity with respect to the applied datasets. The proposed meta-model reduces the number of features by selecting only the affected and sufficient features. In addition, in the training phase, the meta-model system selects the results of the best-predicted classifiers to be used in the testing phase.

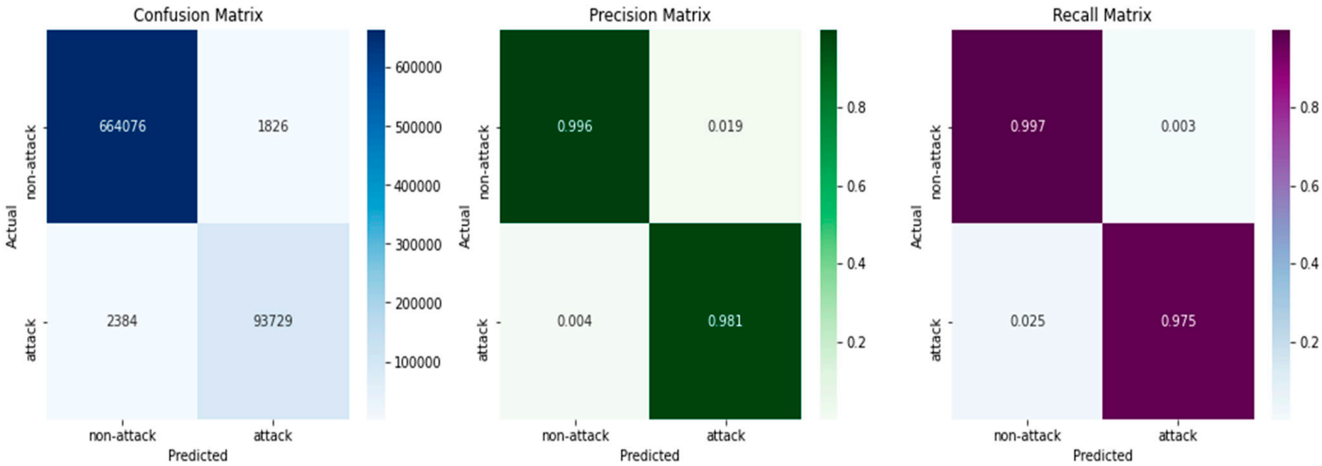

Figure 7. Final measurement matrix when applying meta-model system.

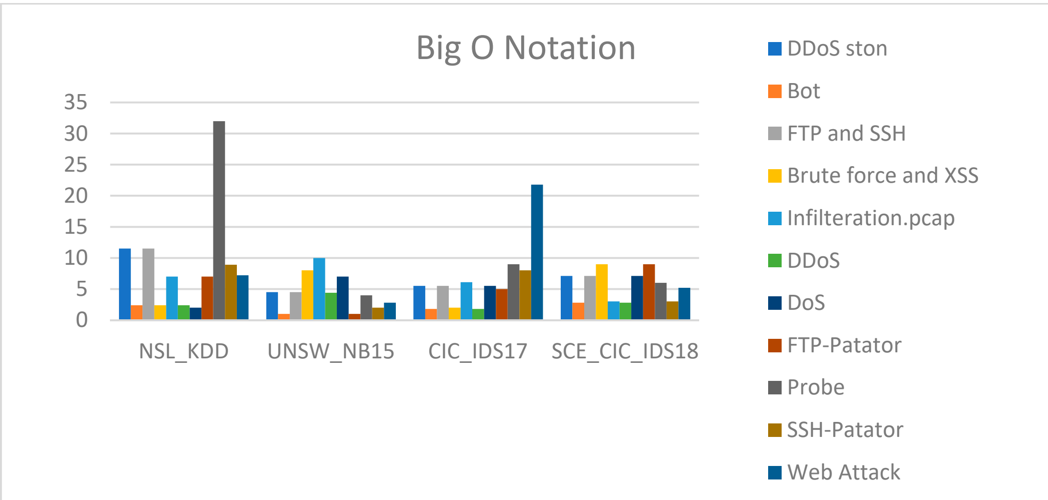

Figure 8. Big O notation idea for four datasets.

3 Analysis Results and Comparison with Other Related Studies

The first stage is very important to clear the datasets and process them from all problems, then pass to the FS stage (chi-square). In this stage, each dataset's class passes through an analysis procedure to check and choose the best effective features' subset to the final results and find the suitable subset feature of NSL_KDD is 20 features, 30 features of the UNSW_NB15, 35-features in CIC_IDS17, and 38-features in SCE_CIC_IDS18. Afterby, the ML and voting techniques stages begin to make each classifier work independently and aggregated applying the voting average technique to return the best result for the classifiers.

The proposal is assessed and compared to other previous systems by accuracy, FAR, DR, and a number of FS, Table 3 demonstrates the outperform of the meta-model is 99% for training and 90.1% for testing, as compared with other similar studies.

4 Challenges

Experimental results indicate that IDS based on a new NIDS is proposed using a meta-model (ML) with DT as a voting technique. The main objective is to build a secure system which able to distinguish malicious/suspicious traffic activities. The proposed meta-model proves sufficiency and effectiveness to detect intrusions and suspicious traffic activities, however, some limitations have come into view to be recommended to other researchers. It includes the following constraints:

- The accuracy of the entire system depends on the average accuracy of all the classes. Hence, for more efficient and accurate results, it is recommended to compute the accuracy of each class a side and accordingly the system average accuracy of all the classes for optimal performance.
- The meta-model system outperforms excellent performance when testing the system by four different datasets, however, it does not consider further attacks sourced by external networks.
- Analyzing data connections aids in the detection of non-detectable attacks throughout the application of IDS to each connection record separately. Thus, it always requires updated preprocessing and FS for accurate analyses.
- Deploying the proposed NIDS to the classified information servers of security establishments. Hence, this requires constant development for up-to-date NIDSs.

Table 3. Results comparison with other studies.

References/ Published Year	Dataset	FS Method	Number of FS	Classification Method	Accuracy %	DR %	FAR %
[30], 2016	NSL_KDD	DT	N/A	EL Methods (Rule base)	80	81	N/A
[31], 2017		KNN	16	NB	83	82	4.83
[32], 2021		symmetrical uncertainty, Information Gain and CFS	32	Gradient Adaptive Rate	85	N/A	15.00
			10		78	N/A	1.00
[33], 2021		Entropy	42	SVM	95	96	5.11
[34], 2021		Wrapper based GA	13	logistic regression as an EL algorithm	97.99	96.64	N/A
	UNSW_NB2015		8		98.73	98.93	N/A
	CIC_ID17		11		98.99	98.75	N/A
[35], 2022	NSL_KDD	Deep NN	N/A	Gradient Boosting algorithm	99	N/A	N/A
[35], 2022	CIC_ID17		N/A		92	N/A	N/A
[11], 2022	NSL_KDD	CFS-RF	30	Voting (RF, and SVM)	99.4	99.9	0.004
	UNSW_NB15		35		99.8	99.6	0.008
	CIC_ID17		40		99.7	99.4	0.0012
Meta-model	NSL_KDD	Chi-square	20	ML with meta-model classifiers (i.e., XGB (C1), Random Forest (C2), DT (C3), AdaBoost (C4), GradientBoosting (C5), LightGBM (C6), and CatBoost (C7)).	99.9	99	0.002
	UNSW_NB15		30		99.5	99	0.004
	CIC_ID17		35		99.8	99	0.0013
	SCE_CIC_IDS18		38		99.3	99	0.0021

4. Conclusions

In nutshell, it was discovered that the existing IDSs are still ineffectual despite having intentionally utilized a range of ML techniques to increase their performance, principally as a result susceptibility of to the anticipated 6G wireless paradigm and the rapidly evolving sophisticated threats. The meta-model system initiated a new IDS mechanism to apply to unbalanced/high dimensional network traffic having a low DR given the needed ML classifiers and voting mechanisms. The proposed meta-model system complexity was reduced while applying Chi-Square to present (20, 30, 35, and 38) features for NSL KDD, UNSW NB15, CIC IDS17, and SCI CIC IDS18, respectively to acquire the ideal subset of the best FS and dimensionality reduction. For each dataset, the experiment's results of the meta-model achieve high accuracies for all datasets reach 0.99% and low FAR values for NSL KDD, UNSW NB15, CIC IDS17, and SCI CIC IDS18 were 0.002, 0.004, 0.0013, and 0.0021, respectively. Other findings are concisely displayed within the results comparison table. The suggested method also outperformed current classification methods. As can be observed, this method significantly increased the IDS market's competitive edge over other strategies. Despite the system's benefits, further work is still required to make it capable of handling potential threats from future infrequent traffic.

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

Funding: This research received no external funding.

Data Availability Statement: NSL_KDD, and UNSW_NB15 Dataset free downloaded from the link: http://www.di.uniba.it/~andresini/datasets.html, accessed on 18 February 2022. CICIDS2017 Dataset free downloaded from the link: http://205.174.165.80/CICDataset/CIC-IDS-2017/Dataset/, accessed on 24 June 2022, and SCE_CIC_IDS18Dataset free downloaded from the link: https://www.unb.ca/cic/datasets/ids-2018.html, accessed on 12 January 2022.

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

References

1. Letaief, K.B.; Chen, W.; Shi, Y.; Zhang, J.; Zhang, Y.J.A. The Roadmap to 6G: AI Empowered Wireless Networks. *IEEE Commun. Mag.* **2019**, *57*, 84–90. [CrossRef]
2. Duan, Z.; Song, P.; Yang, C.; Deng, L.; Jiang, Y.; Deng, F.; Jiang, X.; Chen, Y.; Yang, G.; Ma, Y.; et al. The impact of hyperglycaemic crisis episodes on long-term outcomes for inpatients presenting with acute organ injury: A prospective, multicentre follow-up study. *Front. Endocrinol. (Lausanne)* **2022**, *13*, 1–11. [CrossRef] [PubMed]
3. Lin, Z.; An, K.; Niu, H.; Hu, Y.; Chatzinotas, S.; Zheng, G.; Wang, J. SLNR-based Secure Energy Efficient Beamforming in Multibeam Satellite Systems. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, 1–4. [CrossRef]
4. Lin, Z.; Lin, M.; Wang, J.B.; De Cola, T.; Wang, J. Joint Beamforming and Power Allocation for Satellite-Terrestrial Integrated Networks with Non-Orthogonal Multiple Access. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 657–670. [CrossRef]
5. Mokhtari, S.; Abbaspour, A.; Yen, K.K.; Sargolzaei, A. A machine learning approach for anomaly detection in industrial control systems based on measurement data. *Electron* **2021**, *10*, 407. [CrossRef]
6. Sommestad, T.; Holm, H.; Steinvall, D. Variables influencing the effectiveness of signature-based network intrusion detection systems. *Inf. Secur. J.* **2021**, *31*, 711–728. [CrossRef]
7. Winanto, E.A.; Idris, M.Y.; Stiawan, D.; Nurfatih, M.S. Designing consensus algorithm for collaborative signature-based intrusion detection system. *Indones J. Electr. Eng. Comput. Sci* **2021**, *22*, 485–496. [CrossRef]
8. Creech, G.; Hu, J. A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns. *IEEE Trans. Comput.* **2014**, *63*, 807–819. [CrossRef]
9. Sahu, K.K.; Nayak, S.C.; Behera, H.S. Multi-step-ahead exchange rate forecasting for South Asian countries using multi-verse optimized multiplicative functional link neural networks. *Karbala. Int. J. Mod. Sci.* **2021**, *7*, 7. [CrossRef]
10. Jabardi, M.; Hadi, A.S. Twitter fake account detection and classification using ontological engineering and semantic web rule language. *Karbala. Int. J. Mod. Sci.* **2020**, *6*, 404–413. [CrossRef]
11. Oleiwi, H.W.; Mhawi, D.N.; Al-Raweshidy, H. MLTs-ADCNs: Machine Learning Techniques for Anomaly Detection in Communication Networks. *IEEE Access* **2022**, *10*, 91006–91017. [CrossRef]
12. Oleiwi, H.W.; Al-Raweshidy, H. Cooperative SWIPT THz-NOMA/6G Performance Analysis. *Electronics* **2022**, *11*, 873. [CrossRef]
13. Oleiwi, H.W.; Saeed, N.; Al-Raweshidy, H.S. A Cooperative SWIPT-Hybrid-NOMA Pairing Scheme Considering SIC Imperfection for THz Communications. In Proceedings of the 2022 IEEE 4th Glob Power, Energy Commun Conf GPECOM 2022, Cappadocia, Turkey, 14–17 June 2022; pp. 638–643. [CrossRef]
14. Oleiwi, H.W.; Al-Raweshidy, H. SWIPT-Pairing Mechanism for Channel-Aware Cooperative H-NOMA in 6G Terahertz Communications. *Sensors* **2022**, *22*, 6200. [CrossRef] [PubMed]
15. Oleiwi, H.W.; Saeed, N.; Al-Raweshidy, H. Cooperative SWIPT MIMO-NOMA for Reliable THz 6G Communications. *Network* **2022**, *2*, 257–269. [CrossRef]
16. Zhang, J.; Su, Q.; Tang, B.; Wang, C.; Li, Y. DPSNet: Multitask Learning Using Geometry Reasoning for Scene Depth and Semantics. *IEEE Trans. Neural. Netw. Learn. Syst.* **2021**, 1–12. [CrossRef] [PubMed]
17. Gui, G.; Liu, M.; Tang, F.; Kato, N.; Adachi, F. 6G: Opening New Horizons for Integration of Comfort, Security, and Intelligence. *IEEE Wirel. Commun.* **2020**, *27*, 126–132. [CrossRef]
18. Rajagopal, S.; Kundapur, P.P.; Hareesha, K.S. A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets. *Secur. Commun. Netw.* **2020**, *2020*, 1–9. [CrossRef]
19. Mhawi, D.N.; Oleiwi, H.W.; Saeed, N.H.; Al-Taie, H.L. An Efficient Information Retrieval System Using Evolutionary Algorithms. *Network* **2022**, *2*, 583–605. [CrossRef]
20. Doaa Nteesha Mhawi, A.K. Information Retrieval Using Modified Genetic Algorithm. *Al. Mansour. J.* **2017**, *27*, 15–35. [CrossRef]
21. Oleiwi, H.W.; Saeed, N.; Al-taie, H.L.; Mhawi, D.N. Evaluation of Differentiated Services Policies in Multihomed Networks Based on an Interface-Selection Mechanism. *Sustainability* **2022**, *14*, 3235. [CrossRef]
22. Ghindawi, I.W.; Kadhm, M.S.; Mhawi, D.N. The Weighted Feature Selection Method. *J. Coll. Educ.* **2018**, *3*, 1–12.

23. Mhawi, D.N.; Aldallal, A.; Hassan, S. Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems. *Symmetry* **2022**, *14*, 1461. [CrossRef]
24. Oleiwi, H.; Saeed, N.; Al-Taie, H.; Nteesha, D. An Enhanced Interface Selectivity Technique to Improve the QoS for the Multi-homed Node. *Eng. Technol. J.* **2022**, *40*, 101–109. [CrossRef]
25. Mhawi, D.N. Proposed Hybrid Correlation Feature Selection Forest Panalized Attribute Approach to advance IDSs. *Mod. Sci.* **2021**, *7*, 15. [CrossRef]
26. Hota, H.S.; Shrivas, A.K. Decision Tree Techniques Applied on NSL-KDD Data and Its Comparison with Various Feature Selection Techniques. In *Smart Innovation, Systems and Technologies*; Springer: Cham, Switzerland, 2014; Volume 27, pp. 205–212.
27. Khammassi, C.; Krichen, S. A GA-LR wrapper approach for feature selection in network intrusion detection. *Comput. Secur.* **2017**, *70*, 255–277. [CrossRef]
28. Moon, S.-H.; Kim, Y.-H. An improved forecast of precipitation type using correlation-based feature selection and multinomial logistic regression. *Atmos. Res.* **2020**, *240*, 104928. [CrossRef]
29. Loey, M.; Manogaran, G.; Taha, M.H.N.; Khalifa, N.E.M. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement* **2020**, *167*, 108288. [CrossRef]
30. Gaikwad, D.; Thool, R. DAREnsemble: Decision tree and rule learner based ensemble for network intrusion detection system. *Proc. Smart Innov. Syst. Technol.* **2016**, *50*, 185–193. [CrossRef]
31. Pajouh, H.H.; Dastghaibyfard, G.; Hashemi, S. Two-tier network anomaly detection model: A machine learning approach. *J. Intell. Inf. Syst.* **2015**, *48*, 61–74. [CrossRef]
32. Kanakarajan, N.K.; Muniasamy, K. Improving the accuracy of intrusion detection using gar-forest with feature selection. *Proc. Adv. Intell. Syst. Comput.* **2016**, *404*, 539–547.
33. Mittal, M.; de Prado, R.; Kawai, Y.; Nakajima, S.; Muñoz-Expósito, J. Machine Learning Techniques for Energy Efficiency and Anomaly Detection in Hybrid Wireless Sensor Networks. *Energies* **2021**, *14*, 3125. [CrossRef]
34. Jaw, E.; Wang, X. Feature Selection and Ensemble-Based Intrusion Detection System: An Efficient and Comprehensive Approach. *Symmetry* **2021**, *13*, 1764. [CrossRef]
35. Gupta, N.; Jindal, V.; Bedi, P. CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. *Comput. Secur.* **2022**, *112*, 102499. [CrossRef]