Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.029126
Article

Tech Science Press

Perspicacious Apprehension of HDTbNB Algorithm Opposed to Security Contravention

Shyla[1,*] and Vishal Bhatnagar[2]

[1]NSUT East Campus (Formerly Ambedkar Institute of Advanced Communication Technologies and Research), Guru Gobind Singh Indraprastha University, New Delhi, 110031, India
[2]NSUT East Campus (Formerly Ambedkar Institute of Advanced Communication Technologies and Research), New Delhi, 110031, India
*Corresponding Author: Shyla. Email: shylasinghit@gmail.com
Received: 25 February 2022; Accepted: 21 April 2022

Abstract: The exponential pace of the spread of the digital world has served as one of the assisting forces to generate an enormous amount of information flowing over the network. The data will always remain under the threat of technological suffering where intruders and hackers consistently try to breach the security systems by gaining personal information insights. In this paper, the authors proposed the HDTbNB (Hybrid Decision Tree-based Naïve Bayes) algorithm to find the essential features without data scaling to maximize the model's performance by reducing the false alarm rate and training period to reduce zero frequency with enhanced accuracy of IDS (Intrusion Detection System) and to further analyze the performance execution of distinct machine learning algorithms as Naïve Bayes, Decision Tree, K-Nearest Neighbors and Logistic Regression over KDD 99 dataset. The performance of algorithm is evaluated by making a comparative analysis of computed parameters as accuracy, macro average, and weighted average. The findings were concluded as a percentage increase in accuracy, precision, sensitivity, specificity, and a decrease in misclassification as 9.3%, 6.4%, 12.5%, 5.2% and 81%.

Keywords: Naïve bayes; decision tree; k-nearest neighbors; logistic regression; neighbors classifier

1 Introduction

The rapid movement towards the digital world inculcates the flow of information within the wired and wireless networks spread across the local, metropolitan areas, and wide-area networks around the globe. The technological advancement in communication led to the issue of information security. The hackers and intruders consistently try to breach the critical constituents as availability, integrity, authenticity, and confidentiality. The attempt to deploy intrusions over the network to breach information security is mitigated by using IDS (Intrusion Detection System). IDS system observes malicious activities by keeping track of all outgoing and incoming data packets to identify signatures. The IDS, based on its dynamic functionality, is divided into three categories misuse-based IDS, signature-based IDS, and anomaly-based IDS to track anomalous behavior [1].

The IDS systems generate alarms in case of identification of malicious activities by signature matching, where the signatures of the attacks are matched with the already existing attack database. Immediately after identifying the intrusion, the alarm gets activated by sending a notification to the user with higher accuracy and minimum false positives. The attack database identified the known attacks, but the newly emerging attacks that consistently change the attack patterns are determined by analyzing intrusions behavior [2].

The strange and newly emerging intrusion behavior is matched with the usual network traffic behavior to identify a difference in network behavior for detecting attacks. If the system cannot observe the flow of network data traffic, then the technique of misuse detection is merged with signature-based detection, where the communication protocols are traced [3–5]. The conventional intrusion detection approaches, such as firewalls and anti-intrusion models, were designed by using the architecture of IDS.

The comprehensive learning technology tracks the intrusions by using traditional IDS techniques [6]. In this paper authors comparatively analyze the experimental enactment of different classification learning algorithms such as Naïve Bayes, Decision Tree, K-Nearest Neighbors, and Logistic Regression over the KDD 99 dataset with the proposed HDTbNB (Hybrid Decision Tree-based Naïve Bayes) algorithm. The performance analysis is designed by computing various performance specifications. It is found that if the overall accuracy of the algorithm is high, the false alarm rate is also high. That algorithm is not the best fit for the identification of emerging attacks. The authors proposed an HDTbNB algorithm to mitigate the occurrences of false alarm rates by increasing the performance of IDS. The methodology adopted by the authors focuses on the different research question, which is given as follows:

- RQ1—To find the essential features without data scaling to maximize the model's performance.
- RQ2—Find the performance matrices depending on methodology, parameters, and independent attributes.
- RQ3—To depict a methodology by reducing the false alarm rate and training period to reduce zero frequency with enhanced accuracy.

The paper is designed by incorporating various sections, as Section 2 constitutes related work of different authors with gap analysis. The exploratory analysis of data with its description is defined in Section 3, and Section 4 shows the detailed schema of the proposed algorithm. Section 5 shows the experimental study with the report, Section 6 as performance analysis, and Section 7 infers the proposed work and future scope.

2 Related Work

The flow of information from one node to another within a wired and wireless network makes the data vulnerable to security threats. The Information security system rigorously adapts methodologies to protect data by assuring authorization breaches, disclosure of information, disruption, and destruction. Information security incidents are minimized by inculcating discoveries under the threat management system by considering basic principles of information authentication, confidentially, integrity throughout the system, and non-reputation schemes. The IDS is used in networks where the information is at a very high-security threat of intruders. The primary aim of the authors [7] is to use blockchain technology with IDS as IDSwBC, which is the first of its kind using blockchain technology. The IDS system is designed into two phases signature creation and anomaly detection. The evaluation of the system is measured based on throughput, execution time, processing time, and latency. It is found that the accuracy of a system using hyperledger fabric as blockchain is 97.8% and without using IDSwBC is 97.8%. The machine learning algorithms were analyzed to detect intrusions by authors [8] using feature selection methodology over test and training samples of the UNSW NB-15 dataset. The authors used a chi-squared feature selection filter based algorithm to amplify the performance of IDS. The accuracy parameter attainment of machine learning algorithms such as LR, NB, RF, SGD, and KNN is analyzed by computing the accuracy as 98.42%, 76.59%, 99.57%, 98.16%, and 98.28%. The authors compared the performance of different algorithms over 23 features of dataset which is dissected into two parts training set and testing set and found that the KNN classifier shows the highest accuracy of 99.57%. The work can be extended with multiclass classification.

Intrusions degraded the capability of wireless sensor networks. The authors [9] proposed a GWOSVM-IDS system, the support vector machine learning algorithm with a grey wolf optimizer having 3, 5, and 7 wolves to track the intrusions. The primary aim is to enhance the system's performance by increasing accuracy with fewer false alarms and a high intrusion detection rate with reduced processing time. It is found that the accuracy of the proposed algorithm with 3, 5, and 7 wolves is 79%, 92%, and 96%. The advantage of proposed algorithm is to increase detection rate with decrease in processing time in wireless sensor networks. The challenge of emerging security threats in the communication network allows authors [10] to propose a CTC algorithm based on the HIDS system over the C4.5 detector. The SRRS supervised relative random sampling techniques are used over the multiclass feature selection technique to pre-process a highly imbalanced dataset. It is found that the overall accuracy of 99.6% and 99.5% accuracy is achieved using CICIDS2017 and NSL-KDD datasets. The information access to legitimate intruders is restricted to make the network more secure by [11] authors by working on deep learning algorithms over largescale data in cybernetwork. The authors proposed OCNN-HMLSTM, an optimized neural network, and hierarchical multiscale LSTM by learning spatial-temporal features for hyperparameter optimization. It is found that the OCNN-HMLSTM model achieves an overall accuracy of 90% with a reduced false alarm rate and better coefficient classification.

The authors estimate the healthcare expenditure for medical care [12] depending on body mass index, obesity, aging, and genetic diseases. The public dataset is used to study the impact of healthcare on medical expenses. A linear regression model performance is compared with other algorithms to predict the costs. It is found that the linear regression model shows an accuracy of 97.89%. The authors diagnosed the erratic disease as arrhythmia [13] using TERMAs and FFT algorithms to evaluate ECG signals to denoise R, P, and T signals.

The software defined network is secured from DDoS attacks in which multiple intruder systems target particular server for the purpose of data breach. The information is secured from DDoS attack by using machine learnings algorithms to detect malicious network traffic. The KDD99 dataset is used by authors [14] to train and test the data using SVM and decision tree algorithm and it is found that SVM has higher precision rate [15]. It is found that the performance of IDS is increased by reduction of false alarm rate to provide the security to Unmanned Ariel Vehicles (UAV) network systems. The DRL-BWO algorithm is used by authors with an accuracy of 98.9%.

Tab. 1 shows the gap analysis for different research findings. The authors used different techniques on different datasets where the missing issues are addressed in the table.

Table 1: Gap analysis

S.No.	Dataset used	Technique used	Gap findings
1	UNSW-NB15	Filter based feature selection technique.	The tree's structure changes enormously with the change in parameters due to extensively sensitive tree reproducibility [8].
2	NSL KDD 99	Grey wolf optimizer and SVM.	The algorithm's prediction is improved by creating multiple decision trees depending on the requirement [9].

(Continued)

Table 1 (continued)

S.No.	Dataset used	Technique used	Gap findings
3	NSL-KDD CICIDS2017	Supervised relative random sampling	The induction and pruning algorithms are used to reduce the cost and complexity of space and time for sorting each node at each level [10].
4	NSL-KDD UNSW-NB15 ISCX-IDS	Optimized CNN with multiscale LSTM	The tree supports categorical and continuous variables without normalization and standardization [11].
5	SDN	Decision tree and (Support vector Machine) SVM	The assumption of independent attributes as predictors makes prediction difficult with mutually independent data [14].
6	UAV Networks	Deep Reinforcement learning	If the training data cannot observe the categorized variable as in testing data, then the issue of zero frequency arises [15].

3 Exploratory Analysis

The prior objective is to design the algorithm for intrusion detection learning to differentiate between normal and attacked connections. The KDD99 open-access dataset is used for intrusion detection. Lincoln labs launched the evaluation program of DARPA for intrusion detection with the scope of the survey and experimental-based research over the IDS [16]. The labs were set up to acquire various attacks by simulating the military data over the LAN, creating a false environment. The data flowing over the TCP was tracked over the weeks, and the tracked information shows multiple attack patterns. The majority of attacks fall under the categories labelled as normal and attack. The qualitative and quantitative features were traced over each TCP/IP connection of 100 bytes.

Fig. 1 shows that the dataset is divided into two categories of tracked connection as Normal and Attack. The pie chart shows the distribution of classes where 13449 were regular connections, and multiple intruders attacked 11743.

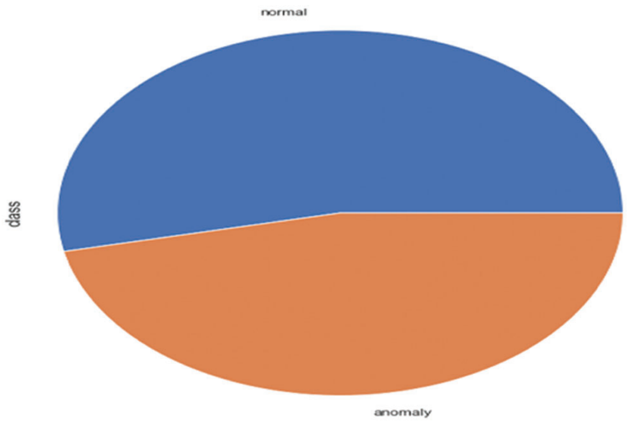

Figure 1: Attack categories

Fig. 2 shows the distribution of established connections over different protocol types. The maximum number of connections established over TCP protocol type and the minimum over ICMP protocol type. The data packets flow using an IP address by establishing two-way communication using a well-defined sequence of flow.

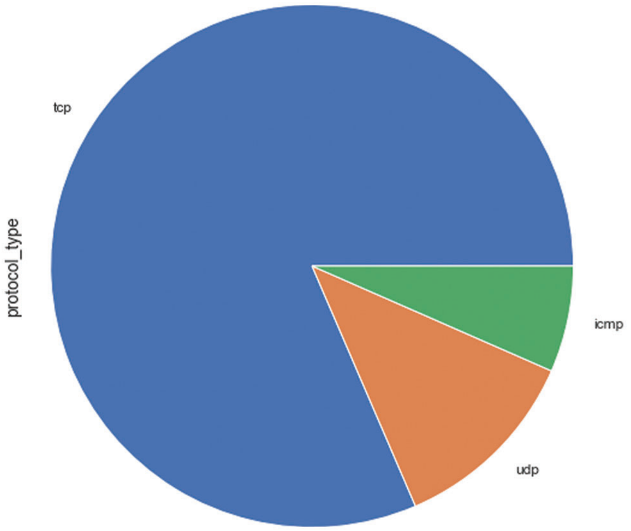

Figure 2: Protocol type

Fig. 3 shows the derived and vital features based on which the normal and attacked categories of TCP/IP connection are further divided. The feature gives the information about the connections, where the total time duration for the connection is established depending upon the type of protocol used by connections, network services used, bytes transferred, error-status using flag, packet size, failed packet transmission, dropped packets, error rate and failed connections.

4 Proposed Methodology

The HDTbNB algorithm is proposed by hybridization of Naïve Bayes and Decision Tree Algorithm. The Naïve Bayes algorithm is a supervised learning approach based on the Bayes Theorem" depending on the naïve theory of independency of pairs of features. The relationship between class variables s defines the Bayes theorem' and feature vectors t_1 and t_2 in Eq. (1)–(5), [17,18].

$$P(s|t_{1,}\, t_2 \ldots\ldots, \;\; t_m) = \frac{P(s)P(t_1, \; t_2 \ldots\ldots, \;\; t_m|s)}{P(t_1, \; t_2 \;\; \ldots\ldots, \;\; t_m)} \tag{1}$$

Considering the naïve theory of independency,

$$P(t_j|s, \;\; t_1, \;\; \ldots\ldots, \;\; t, \;\; t_{i+1} \ldots \;, \;\; t_m) = P\left(\frac{t_m}{s}\right) \tag{2}$$

For all, the value of m relationship is defined as,

$$P(s|t_1, \;\; \ldots\ldots, \;\; t_b) = \frac{P(s)\prod_{m-1}^{b} \; P(t_m|s)}{P(t_1, \; \ldots\ldots, \; t_b)} \tag{3}$$

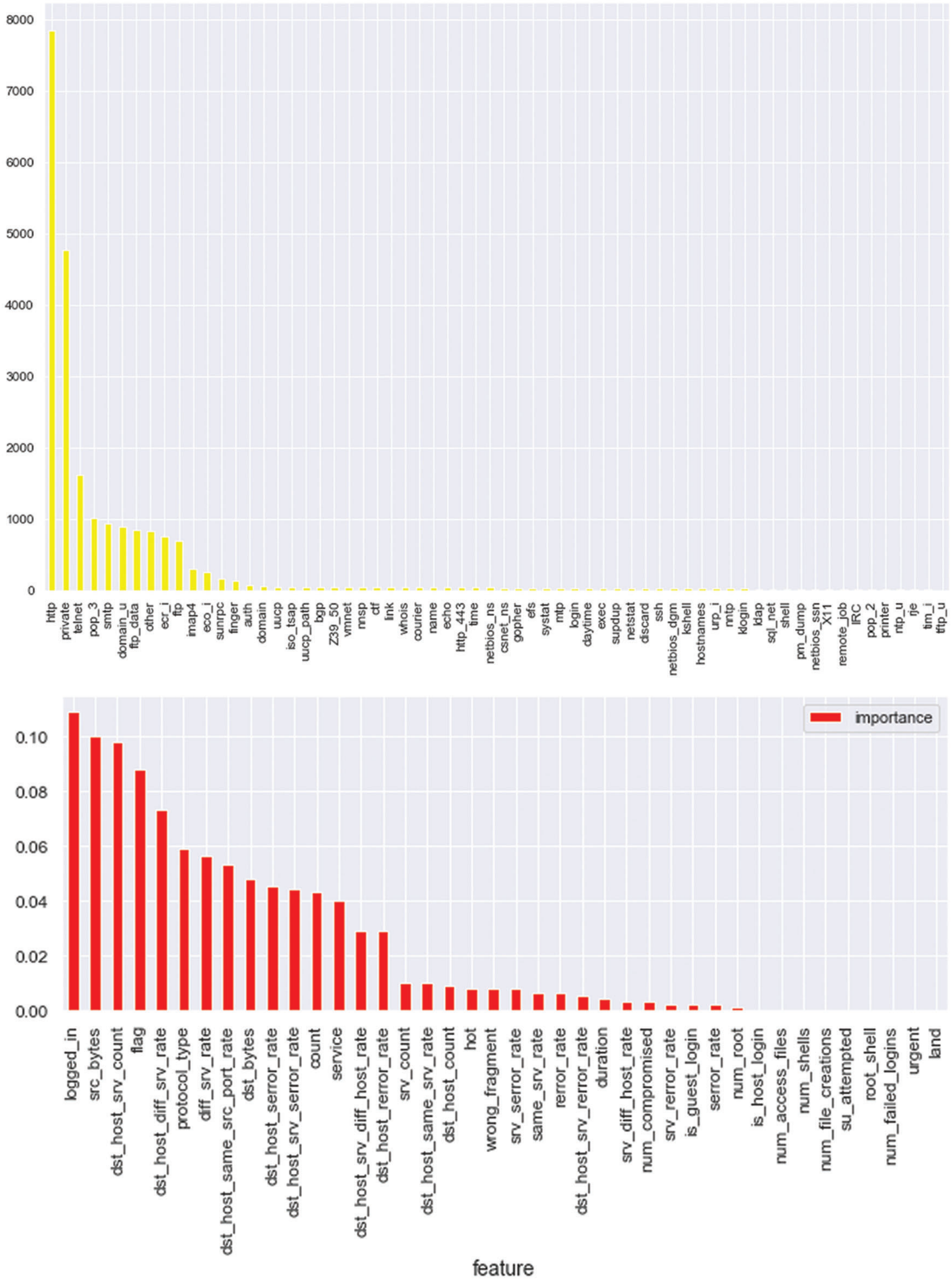

Figure 3: Derived and important features

The classification theorem is defined as,

$$P(s|t_1,\ \ldots,\ \ t_b)\alpha P(s)\Pi_{n=1}^{b}P\left(\frac{t_n}{s}\right) \tag{4}$$

$$\hat{s} = \text{ar } maximumP(s)\ \Pi_{n=1}^{b}P(t_n|s) \tag{5}$$

The MAP (Maximum of a Posteriori) is used to estimate P(s) and P(t_n | s); this presents the relative frequency for s class of training set in Algorithm 1.

Algorithm 1:

```
Begin:
Here u represents the dataset, the count of document c as p, the categorized classes as l, and the number of terms is q.
    Measure data length as len (u)
        for every l $\varepsilon$ $L$
            continue
                count of a document as (c, l)
                P [l] ← $p_l$ \ p
                Merge the text data from classes in the document
            if, elements in the set
        for every q $\varepsilon$ u
            continue
                do $q_{lt}$ ← compute count of text terms ($Txt_l$, q)
            if, total counted terms
        for every q $\varepsilon$ $u$
            continue
                while conprobab[t][l] ← [len(ud)/2–1]
                if achieved, conditional probability
            return u, p, conprobab
terminate.
```

In Algorithm 1 the assumptions were used for the distribution of P(x_i | y), which differs from different naïve bayes classifiers. The naive bayes is also classified as a descent classifier. The one-dimensional distribution is computed by estimating each distribution independently for decoupling of conditional features of classes. The model is designed using the aspects of decision tree modeling, that is, induction of tree and tree pruning. The tree induction takes the input as pre-defined inputs and the process of splitting the dataset for categorization into nodes and sub-nodes. Pruning is the process of removing sub-nodes which is the reverse of the induction process.

Fig. 4 shows the flow chart for the proposed methodology which is the hybridization of decision tree and naïve bayes algorithm. The algorithm is used to take the input for maximization of objective function based on profit. The outcomes of the classification determines the output depending on data instances. The nodes in tree represents the decision based on given input by moving to the next node. The data is dissected into two parts as training set and testing set to find that the model is not overfitted. The each and every attribute of data is considered for the examination at every node after traversal. The values in list is considered by categorization into classes using probability. The attribute is evaluated based on list and selection of categorical attribute.

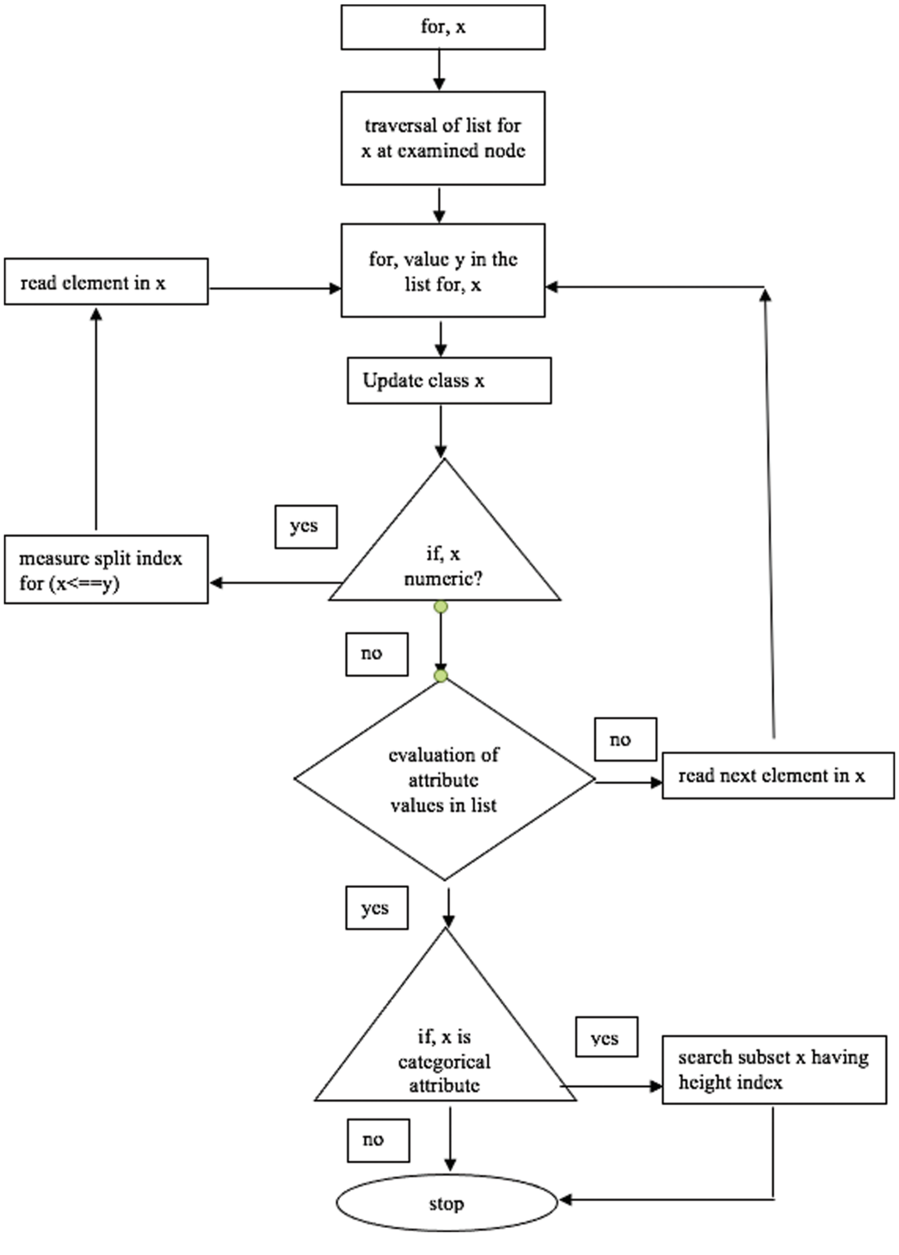
for, x
traversal of list for x at examined node
for, value y in the list for, x
read element in x
Update class x
yes
if, x numeric?
measure split index for (x<==y)
no
evaluation of attribute values in list
no
read next element in x
yes
if, x is categorical attribute
yes
search subset x having height index
no
stop

Figure 4: HDTbNB flow chart

The selection of nodes, sub-nodes, root nodes, and terminal nodes from a dataset consisting of N instances is made by following criteria of computing Entropy, Information gain, Ginni index, Gain ratio, Reduction invariance, and Chi-Square. The value for each model is computed by following the requirements and is placed in the tree.

The Entropy for single attributes is computed in Eq. (6):

$$E(cs) = \sum_{i=1}^{cs} -q_j log_2 q_j \tag{6}$$

where CS is the Current State, q_j is the probability of any event j of state CS or percentage of class j in a node of state CS.

The Entropy for multiple attributes is given in Eq. (7), [19]:

$$E(CS,\ Y) = \sum_{CS \in Y} P(CS)E(CS) \tag{7}$$

Here CS is the current state, and Y is the selected attribute,

Algorithm 2:

```
Begin: Here, P is the classification sample data.
if P $\neq null$, numerical_instances > 0
    create tree
        continue
            maximum I_Gain ← 0
            split_S ← null
            i ← Entropy of instances
            if all instances s in p do
                gain ← IGain(s, i)
                if gain(g) > maximumGain then
                    maximumGain ← gain(g)
                    split_S ← s
                terminate if
            terminate for
            T(p, splitS)
        If all the T is processed
    terminate the procedure
Here M is the unpruned data,
    Tree pruning procedure
        if all r in m, then
            for leaf nodes l of r then
                the replacement of r with c, then
                    reclassification of nodes r
                terminate if
            terminate for
        terminate if
    end
```

In Algorithm 2 the Information gain is computed to determine how precisely the instances split the training samples according to classified targets. The attribute having high information gain and small Entropy is considered. The information gain is defined in Eqs. (8), (9), [20]:

$$IGain(CS,\ Y) = E(CS) - E(CS,\ Y) \tag{8}$$

$$IGain = E(b) - \sum_{n=1}^{m} E(n,\ a) \tag{9}$$

Here are instances of the dataset before splitting, m is the total number of sub-sets created by separating, and (n, a) represents the subset n created after splitting.

The Gini Index is defined as a cost function measured by subtracting each squared probability by one to determine the dataset splits in Eqs. (10), (11), [20].

$$G = 1 - \sum_{j=1}^{cs} (q_j)^2 \tag{10}$$

$$GRatio = \frac{IGain}{SpInfo} = \frac{E(b) - \sum_{i=1}^{m} E\ (i,\ a)}{\sum_{i=1}^{m} v_i log_2 v_i} \tag{11}$$

The gain ratio is used over information gain to reduce the bias of selecting many attributes with different values. It is obtained by measuring Information gain based on split information. The dataset is spliced using information gain for other attributes in Eq. (12), [20].

$$var = \frac{\sum (y - \bar{y})^2}{m} \tag{12}$$

The best split is selected by calculating the variance for each node, where each division has a weighted average to reduce the regression problem. The criteria are used for splitting the population with low variance.

The quantitative security analysis of the proposed methodology is focused by addressing the interdependence between intruder constraints and attack effectiveness. The centroidal quantitative analysis is implicated over anomaly detection.

If the dataset is, D = {d1,……..,d_m}, where d belongs to the same distribution of D.

For centroidal quantitative analysis, The Euclidean distance methodology used for measuring anomaly as,

$$func(D) = ||d - \frac{1}{m} \sum_{j=1}^{m} d_i\ || \tag{13}$$

The performance of the proposed algorithm is analysed in the experimental analysis section based on accuracy, precision, recall, and F1_Scoreparameters with the comparative analysis.

5 Experimental Analysis

The different classifier model algorithms as Naïve Bayes, Decision Tree, Neighbors, Logistic Regression and HNBDT were applied over the training and testing dataset to evaluate and validate the performance of learning models for the prediction of intrusions based on tracked signatures. The classification learning model is estimated by determining the Accuracy, Macro Average and Weighted Average as performance matric.

"The model computes Accuracy, Precision, Recall, and F1-Score by calculating different weigh as true positive (t_p), true negative (t_n), false positive (f_p), and false negative (f_n)".The precision is given by the number of observation that is positive and is predicted to be positive. The recall shows the ability of the model to find observations that were actually positive randomly. F1-Score is the harmonic mean of recall and precision used to evaluate the complete correctness to form a positively predicted environment in Eqs. (13)–(16), [21,22].

$$Accuracy = \frac{t_p + t_n}{t_p + f_p + t_n + f_n} \tag{14}$$

$$Precision = \frac{t_p}{t_p + f_p} \tag{15}$$

$$Recall = \frac{t_p}{t_p + f_n} \tag{16}$$

$$F1\_Score = \frac{2 * Pre * Rec}{Pre + Rec} \tag{17}$$

Tab. 2 shows the performance metrics for model evaluation by measuring different parameters such as Accuracy, Precision, Recall, and F1-Score. The other parameters, Macro Average and Weighted average, are calculated to evaluate multiclass classifiers by aggregating and averaging the evaluation.

Table 2: Model evaluation

Naïve bayes classifier model evaluation

	Precision	Recall	F1-Score	Support
Anomaly	0.96	0.85	0.91	8356
Normal	0.87	0.95	0.91	9278
Accuracy			0.90	16725
MacroAvg	0.90	0.91	0.90	16725
WeightedAvg	0.90	0.92	0.90	16725

Decision tree classifier model evaluation.

	Precision	Recall	F1-Score	Support
Anomaly	0.97	0.84	0.94	8356
Normal	0.89	0.95	0.94	9278
Accuracy			0.93	16725
MacroAvg	0.93	0.94	0.94	16725
WeightedAvg	0.93	0.95	0.94	16725

Neighbors classifier model evaluation

	Precision	Recall	F1-Score	Support
Anomaly	0.98	0.97	0.98	8356
Normal	0.99	0.98	0.99	9278
Accuracy			0.98	16725
MacroAvg	0.97	0.96	0.97	16725
WeightedAvg	0.97	0.97	0.98	16725

(Continued)

Table 2 (continued)

LogisticRegression model evaluation				
	Precision	Recall	F1-Score	Support
Anomaly	0.95	0.93	0.94	8356
Normal	0.94	0.96	0.95	9278
Accuracy			0.96	16725
MacroAvg	0.95	0.94	0.96	16725
WeightedAvg	0.94	0.94	0.94	16725

Tab. 3 shows the validation of the evaluated model by using test results of the trained dataset. The performance metric is validated by measuring the scoring parameters and multiclass classifiers.

Table 3: Model validation

Naïve bayes classifier model test results evaluation				
	Precision	Recall	F1_Score	Support
Anomaly	0.95	0.86	0.88	3397
Normal	0.87	0.95	0.90	4059
Accuracy			0.90	7487
MacroAvg	0.92	0.89	0.91	7487
WeightedAvg	0.92	0.90	0.89	7487
Decision tree classifier model test results evaluation				
	Precision	Recall	F1_Score	Support
Anomaly	0.99	0.97	0.97	3397
Normal	0.97	0.98	0.98	4059
Accuracy			0.98	7487
MacroAvg	0.98	0.96	0.97	7487
WeightedAvg	0.96	0.97	0.96	7487
Neighbors classifier model test results evaluation				
	Precision	Recall	F1_Score	Support
Anomaly	0.97	0.97	0.97	3397
Normal	0.98	0.98	0.98	4059
Accuracy			0.97	7487
MacroAvg	0.98	0.98	0.98	7487
WeightedAvg	0.96	0.97	0.98	7487

Table 3 (continued)

LogisticRegression model test results evaluation				
	Precision	Recall	F1_Score	Support
Anomaly	0.95	0.95	0.94	3397
Normal	0.94	0.96	0.95	4059
Accuracy			0.95	7487
MacroAvg	0.95	0.94	0.96	7487
WeightedAvg	0.95	0.95	0.95	7487

Tab. 4 determines the extraction of true positive (t_p), true negative (t_n), false positive (f_p), and false negative (f_n) for different learning algorithms. It is found that there are distinct values for algorithms; by this, it is computed in Tab. 4 which algorithm predicted the anomalous behavior of intruder as normal, the normal behavior of connection as anomalous.

Table 4: Performance parameters

Algorithms	TP	FP	TN	FN
Naïve Bayes Classifier	2979	190	3873	516
Decision Tree Classifier	3480	26	4036	16
Neighbors Classifier	3455	24	4038	41
Logistic Classifier	3295	135	3927	201
HNBDT	3492	10	4048	8

Tab. 5 shows the distinct behavior of algorithms where the naïve Bayes algorithm has the maximum number of false positive and false negative depending on the high false alarm rate; the algorithm determines the anolomous behavior of intruders as normal. The algorithms such as Decision Tree, Neighbors, Logistic Regression" and HNBDT predict the anomalous behavior.

Table 5: Expected and predicted outcomes

Naïve bayes classifier	Expected	Predicted
Decision Tree Classifier	Anomaly	Normal
Neighbors Classifier	Anomaly	Anomaly
Logistic Classifier	Anomaly	Anomaly
HNBDT	Anomaly	Predicted

6 Performance Analysis

The performance pursuance of the proposed algorithm is measured by computing multiclass classification parameters along with performance metric parameters. The HDTbNB algorithm shows the accuracy, precision, recall, F1_score and average.

Tab. 6 determines the performance evaluation of the HDTbNB algorithm with an accuracy of 99%. The algorithm performance is better than other algorithms with high precision and a lower false alarm rate.

Table 6: Performance evaluation

	HNBDT Algorithm			
	Precision	Recall	F1_Score	Support
Anomaly	0.98	0.98	0.98	517
Normal	0.97	0.98	0.99	188
Accuracy			0.99	705
Macro Avg	0.97	0.97	0.98	705
Weighted Avg	0.98	0.96	0.97	705

Fig. 5 shows the true positive (t_p), true negative (t_n), false positive (f_p) and false negative (f_n) for Naïve Bayes, Naïve Bayes with Decision Tree and HDTbNB learning algorithms, where the $t_{p,}$ t_n, f_p and f_n for Naïve Bayes is 2980, 187, 3873 and 518, $t_{p,}$ $t_{n,}$ f_p and f_n for Naïve Bayes with Decision Tree is 505, 14, 176, 10 and $t_{p,}$ t_n, f_p and f_n for HDTbNB is 3492, 13, 4050 and 9.

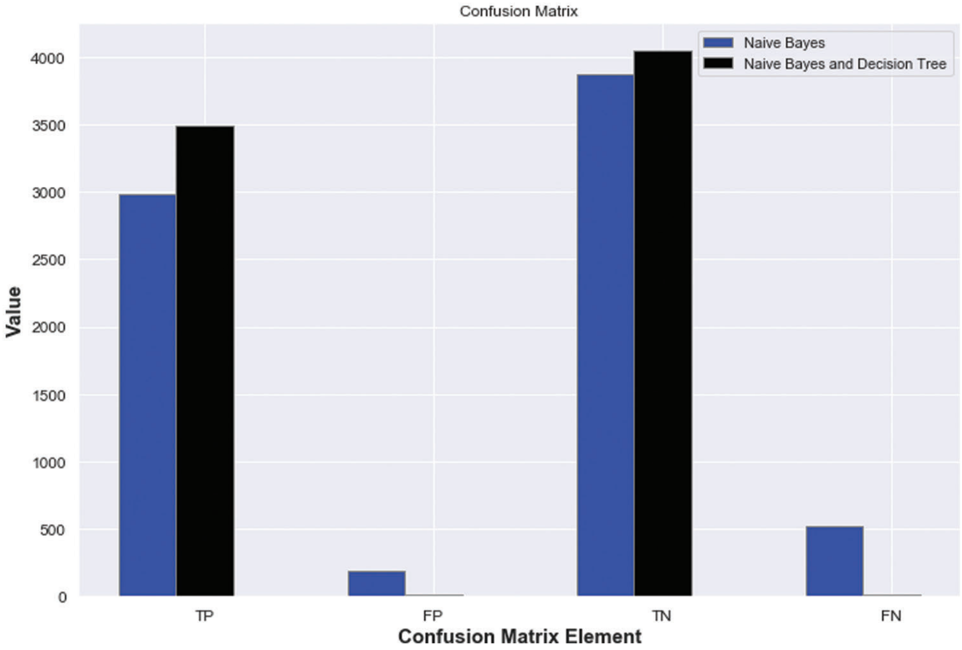

Figure 5: Confusion matrix elements

Fig. 6 shows the comparative performance matrices for comparing accuracy, misclassification, precision, sensitivity and specificity of the Naïve Bayes with Decision Tree and HDTbNB learning algorithm. It is found that the algorithm has increased performance compared to existing algorithms.

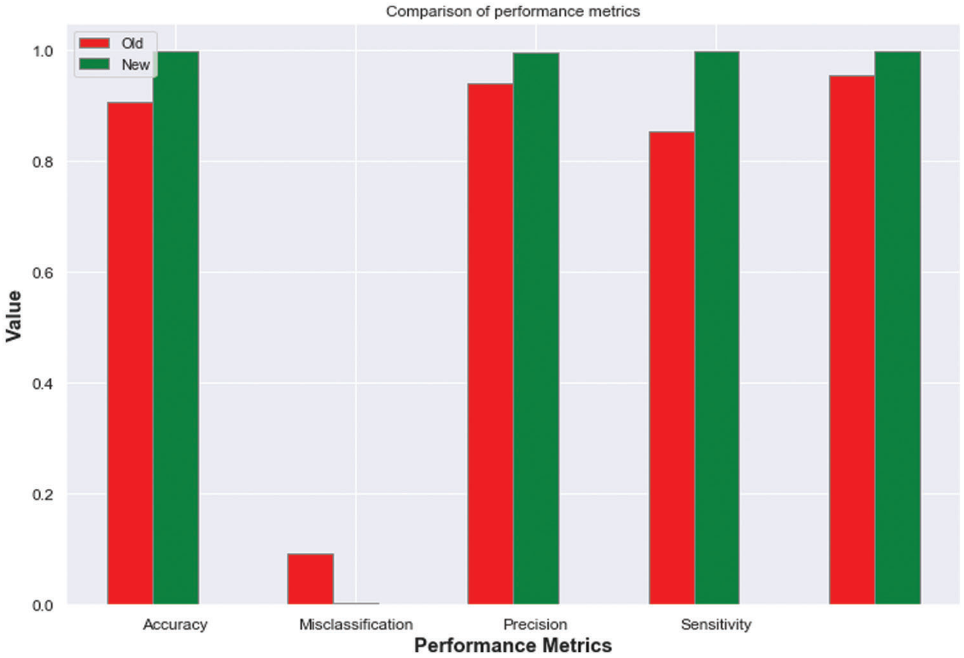

Figure 6: Comparative performance matrices

Tab. 7 shows the percentage increase in the performance of Naïve Bayes with Decision Tree and HDTbNB algorithm for the Kdd99 dataset, and it is found that performance is enhanced by considering accuracy, misclassification, precision, sensitivity and specificity.

Table 7: Percentage increase in performance

Performance evaluation	Naïve bayes with decision tree	HDTbNB	Percentage
Accuracy Increase Percentage	0.905610233742	0.998178066121	9.3%
Misclassification Decrease Percentage	0.084056514025	0.003000712868	81.0%
Precision Increase Percentage	0.930564281895	0.994287835522	6.4%
Sensitivity Increase Percentage	0.872110247867	0.997427101200	12.5%
Specificity Increase Percentage	0.943505571285	0.995798029556	5.2%

Fig. 7 shows the performance of the algorithm depending on percentage increase in accuracy, precision, sensitivity and percentage decrease in misclassification for the proposed algorithm in comparison with naïve bayes and decision tree.

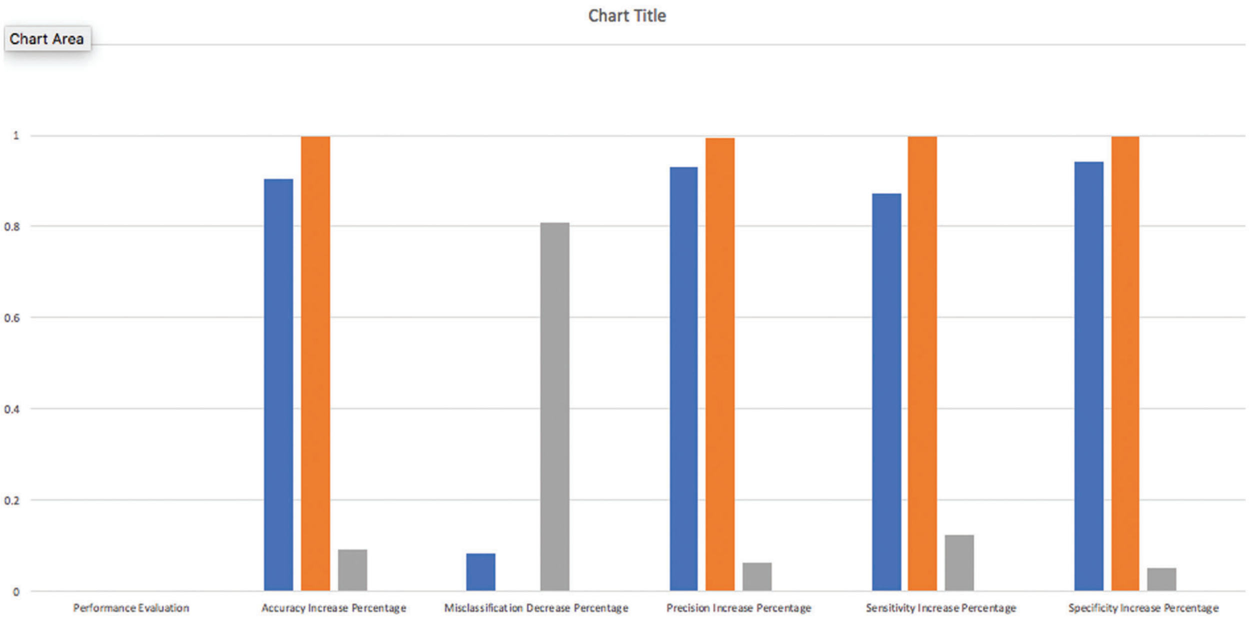

Figure 7: HDTbNB algorithm performance

7 Conclusion and Future Scope

Hybridization of machine learning technology is used to track emerging intrusions by using conventional IDS techniques. In this paper, the author proposed the HDTbNB algorithm by combining decision tree and naïve bayes algorithms to reduce the rate of identified false alarms by increasing the performance of IDS and comparatively analyzing the performance of different machine learning classification algorithms such as Naïve Bayes, Decision Tree K-Nearest Neighbors, and Logistic Regression over KDD 99 dataset with proposed HDTbNB algorithm. The performance analysis is designed by computing various performance specifications. The performance of HDTbNB is evaluated by computing various specifications such as accuracy, macro average, and weighted average. It is found that the proposed algorithm shows the percentage increase in accuracy, precision, sensitivity, and specificity as 9.3%, 6.4%, 12.5%, 5.2%, and a decrease in misclassification to 81.0%. The future scope of the IDS involves the focus on a distinct domain:

- To incorporate Artificial Intelligence (AI) based technology to identify intrusions against cyber security threats in IoT for futuristic smart living.
- To observe the performance of a newly emerging fractal analysis algorithm for intrusion detection in RADAR-based communication systems.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they do not have any conflicts of interest regarding the presented study."

References

[1] R. Zhang, J. P. Condomines and E. Lochin, "A multifractal analysis and machine learning based intrusion detection system with an application in a uas/radar system," *Drones*, vol. 6, no. 1, pp. 21, 2022.

[2] O. Alzahrani and M. J. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, pp. 111, 2021.

[3] S. Smys, A. Basar and H. Wang, “Hybrid intrusion detection system for internet of things (IoT),” *Journal of ISMAC*, vol. 2, no. 4, pp. 190–199, 2020.

[4] S. M. Kasongo and Y. Sun, “A deep learning method with wrapper based feature extraction for wireless intrusion detection system,” *Computers & Security*, vol. 92, pp. 101752, 2020.

[5] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos and P. Burnap, “A supervised intrusion detection system for smart home IoT devices,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042–9053, 2019.

[6] D. Jin, Y. Lu, J. Qin, Z. Cheng and Z. Mao, “Swiftids: Real-time intrusion detection system based on lightgbm and parallel intrusion detection mechanism,” *Computers & Security*, vol. 97, pp. 101984, 2020.

[7] R. S. Khond and V. Ulagamuthalvi, “Blockchain: Secured solution for signature transfer in distributed intrusion detection system,” *Computer Systems Science and Engineering*, vol. 40, no. 1, pp. 37–51, 2022.

[8] G. Kocher and G. Kumar, “Analysis of machine learning algorithms with feature selection for intrusion detection using unsw-nb15 dataset,” *Available at SSRN 3784406*, 2021.

[9] M. Safaldin, M. Otair and L. Abualigah, “Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 1559–1576, 2021.

[10] R. Panigrahi, S. Borah, K. A. Bhoi, F. M. Ijaz and M. Pramanik, “A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets,” *Mathematics*, vol. 9, no. 7, pp. 751, 2021.

[11] R. P. Kanna and P. Santhi, “Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features,” *Knowledge-Based Systems*, vol. 226, pp. 107132, 2021.

[12] I. A. Taloba, A. El-Aziz, M. Rasha, M. H. Alshanbari and H. A. A. El-Bagoury, “Estimation and prediction of hospitalization and medical care costs using regression in machine learning,” *Journal of Healthcare Engineering*, vol. 2022, 2022. https://doi.org/10.1155/2022/7969220.

[13] I. A. Taloba, R. Alanazi, R. O. Shahin, A. Elhadad, A. Abozeid *et al.,* “Machine algorithm for heartbeat monitoring and arrhythmia detection based on ECG systems,” Computational Intelligence and Neuroscience, 2021, https://doi.org/10.1155/2021/7677568.

[14] M. K. Sudar, M. Beulah, P. Deepalakshmi, P. Nagaraj and P. Chinnasamy, “Detection of distributed denial of service attacks in SDN using machine learning techniques,” *Int. Conf. on Computer Communication and Informatics (ICCCI)*, pp. 1–5, 2021.

[15] V. Praveena, A. Vijayaraj, P. Chinnasamy, I. Ali, R. Alroobaea *et al.,* “Optimal deep reinforcement learning for intrusion detection in UAVs,” *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2639–2653, 2021.

[16] D. Dua and C. Graff, “UCI machine learning repository,” CA: University of California, School of Information and Computer Science, 2019, http://archive.ics.uci.edu/ml.

[17] A. M. Khan, “Hcrnnids: Hybrid convolutional recurrent neural network-based network intrusion detection system,” *Processes*, vol. 9, no. 5, pp. 834, 2021.

[18] R. V. Mendonça, A. A. Teodoro, L. R. Rosa, M. Saadi and C. D. Melgarejo, “Intrusion detection system based on fast hierarchical deep convolutional neural network,” *IEEE Access*, vol. 9, pp. 61024–61034, 2021.

[19] L. Yang, A. Moubayed and A. Shami, “Mth-ids: A multitiered hybrid intrusion detection system for internet of vehicles,” *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 616–632, 2021.

[20] H. Madadum and Y. Becerikli, “A Resource-efficient convolutional neural network accelerator using fine-grained logarithmic quantization,” *Intelligent Automation & Soft Computing*, vol. 33, no. 2, pp. 681–695, 2022.

[21] A. S. Kumar, S. Padma and S. Madhubalan, “Distribution network reconfiguration using hybrid optimization technique,” *Intelligent Automation & Soft Computing*, vol. 33, no. 2, pp. 777–789, 2022.

[22] M. R. Saqib, S. A. Khan, Y. Javed, S. Ahmad and K. Nisar, “Analysis and intellectual structure of the multi-factor authentication in information security,” *Intelligent Automation & Soft Computing*, vol. 32, no. 3, pp. 1633–1647, 2022.