The Journal of Supercomputing (2022) 78:17403–17422
https://doi.org/10.1007/s11227-022-04568-3

An investigation and comparison of machine learning approaches for intrusion detection in IoMT network

Adel Binbusayyis[1] · Haya Alaskar[1] · Thavavel Vaiyapuri[1] · M. Dinesh[2]

Accepted: 27 April 2022 / Published online: 18 May 2022

Abstract

Internet of Medical Things (IoMT) is network of interconnected medical devices (smart watches, pace makers, prosthetics, glucometer, etc.), software applications, and health systems and services. IoMT has successfully addressed many old healthcare problems. But it comes with its drawbacks essentially with patient's information privacy and security related issues that comes from IoMT architecture. Using obsolete systems can bring security vulnerabilities and draw attacker's attention emphasizing the need for effective solution to secure and protect the data traffic in IoMT network. Recently, intrusion detection system (IDS) is regarded as an essential security solution for protecting IoMT network. In the past decades, machines learning (ML) algorithms have demonstrated breakthrough results in the field of intrusion detection. Notwithstanding, to our knowledge, there is no work that investigates the power of machines learning algorithms for intrusion detection in IoMT network. This paper aims to fill this gap of knowledge investigating the application of different ML algorithms for intrusion detection in IoMT network. The investigation analysis includes ML algorithms such as *K*-nearest neighbor, Naïve Bayes, support vector machine, artificial neural network and decision tree. The benchmark dataset, Bot-IoT which is publicly available with comprehensive set of attacks was used to train and test the effectiveness of all ML models considered for investigation. Also, we used comprehensive set of evaluation metrics to compare the power of ML algorithms with regard to their detection accuracy for intrusion in IoMT networks. The outcome of the analysis provides a promising path to identify the best the machine learning approach can be used for building effective IDS that can safeguard IoMT network against malicious activities.

Keywords Internet of medical things (IoMT) · Security issues · Intrusion detection system · Machine learning (ML) · *K*-nearest neighbor (KNN) · Naïve Bayes (NB) · Support-vector machine (SVM) · Artificial neural network (ANN) · Decision tree (DT)

Extended author information available on the last page of the article

1 Introduction

The outbreak of coronavirus pandemic (Covid-19) has driven many countries to an adverse situation imposing a serious threat to human life on social interaction [1, 2]. The challenging situation of pandemic demands technological innovation to provide impactful healthcare services for quarantined patients and preclude the spread of infection. In tandem, IoMT is foreseen as a potential solution to enable the doctors maintain social distancing and treat patients remotely which is mandate to confront the ongoing pandemic crisis [3]. The continuous advancement in IoT has accelerated the popularity of IoMT in paving way to build smart healthcare system and serve patients more effectually with fast healthcare services. Many countries have deployed IoMT to promote timely diagnosis through real-time patient monitoring and save their people life during the unpredictable pandemic crisis [4].

IoMT also called as healthcare IoT amalgamates assortment of healthcare devices with healthcare information technology systems to share sensitive patient data with medical experts for personalized medical response [5]. The introduction of IoMT has opened possibilities in healthcare with regard to patient convenience in receiving quality medical services from home. Also, in terms of cost by alleviating unnecessary hospital visits and stays [6]. It has also reduced the stress on health professionals by automating several tasks and transforming hospital-based practices to telehealth practices. All these benefits of IoMT come on the top of significant improvement in precision of diagnosis and accuracy of treatment. Like so, healthcare providers also envision IoMT as a cornerstone for facing the challenge of pandemic crisis to monitor and treat several patients remotely at the same time without requiring extra healthcare facilities [3].

The sharp rise in IoMT popularity with its unparalleled benefits has simultaneously turned out to be the prime and attractive target for cyberattacks [7, 8]. The open environment of IoMT and its design with several vulnerable points has captivated the major attention of cybercriminals to hone their talents in exploiting the IoMT vulnerabilities with more sophisticated cyberattacks to achieve their evil desires. The cyberattacks on IoMT can cause devastating effects and jeopardize the patient life [9]. For example, if an adversary gains control of IoMT devices such as an insulin pump or pacemaker can configure and put the patient life to death [5, 10]. Thus, the Security of IoMT is a major concern on global concern that needs to be carefully addressed at the very first level for its adoption to be effective and continue its upsurge exponentially in the future.

Many security solutions such as firewalls, Antivirus, and IDS have been proposed to safeguard the network resources from intrusions and cyberattacks [10]. Among several security solutions, IDS that monitors network traffic for malicious activities launched from inside and outside the network is regarded as primary and powerful defense mechanism of most organization and has received increased attention in recent years [11]. In general, IDS is categorized into signature-based (SIDS) and anomaly-based IDS (AIDS). SIDS utilizes database with predefined attack patterns to detect attack whether the network traffic is intrusion or not [12].

In case if the attack signature is not found in database, it fails to detect until the database is updated regularly with new attack signatures. On contrary, AIDS detects attack based on network traffic behavior without the need for prior knowledge about the attack signature. Herein, AIDS is preferred and has become the hotspot of research in the field of network security [13].

ML algorithms have received major attention over the past decades as a promising solution for enhancing the detection accuracy of AIDS with application to a wide range of network environments such as cloud, IoT, and Industrial IoT [14, 15]. However, it was surprising to note that the literature is lacking studies that investigate the power of ML algorithms for intrusion detection in IoMT networks. Accordingly, the research work in this paper intends to deepen into this aspect by investigating the application of ML algorithms from different families with regard to their ability to enhance the attack detection accuracy in IoMT networks. In lieu of this, the research work concentrates on analyzing the five most powerful ML algorithms namely NB, KNN, DT, ANN, and SVM. Further, the most recent benchmark intrusion dataset, Bot-IoT which is publicly available with a comprehensive set of sophisticated attack patterns for IoMT network traffic is used to train and test the effectiveness of all ML algorithms considered for investigation. Also, a comprehensive set of evaluation metrics are employed to assess the power of the ML algorithm for gain in accuracy against intrusion detection across IoMT networks. The analysis results provide a promising path to identify the best ML algorithm that can be used for building effective AIDS and safeguarding the IoMT network against malicious activities. The key contribution of the research work is summarized in three points:

(a) Investigate the advantage of applying ML algorithms such as NB, KNN, DT, ANN, and SVM in building AIDS for IoMT network using the recent benchmark intrusion dataset, Bot-IoT.
(b) Compare and assess the effectiveness of considered ML algorithms for gain in detection accuracy through comprehensive set of evaluation metrics.
(c) Analyze and identify the idle ML algorithm based on the obtained comparison results that can be used to build AIDS for securing IoMT network.

2 IoMT system architecture and security issues

This section discusses the IoMT architecture and its security implications. This enables comprehension of the subsequent sections, in which we show the application of machine learning methods to the design of IDS for IoMT systems.

2.1 IoMT system architecture

In the healthcare industry, IoMT is an adaptation of IoT networks that has been tailored for purpose of monitoring a number of various types of vital signs such as blood pressure, glucose level, and EEG. The primary goal of IoMT is to lessen the stress of hospitalization for patients. Allowing the patients to move about the medical and nonmedical surroundings, while their vital signs are constantly monitored with no interruption, is a critical component of providing high-quality medical services. IoMT, as a focused embodiment of IoT in the medical area, adheres to the industry standard three-tier architecture of IoT applications comprising the following layers, viz., perception, network, and transmission layer [16]. The architecture of the IoMT is presented in Fig. 1.

(a) *Perceptual layer* The key responsibility of this layer is to collect patient data from sensor devices and facilitate managing access control to devices. For example, in the patient care systems, several sensors are linked to the patient's body to monitor their state and offer assistance as required.

(b) *Network layer* This layer serves as IoMT system backbone. It leverages the Internet, the mobile communication network, and other public networks to transfer data accurately and reliably. It primarily integrates diverse networks, data formats, and other information. It also constructs a service support platform on top of it, providing an open interface for multiple application layer services.

(c) *Application layer* This layer provides the user interface for managing, controlling, and interacting with IoT devices. For the reason of achieving scalability, integrity, and cost-effectiveness, the healthcare industry has been increasingly migrating to cloud systems. As a result, the chances of sophisticated attacks pose a security challenge to the healthcare industry.

2.2 Security issues

In recent years, the rapid advancement of IoMT and the growing number of medical devices in IoMT, have attracted the attention of cybercriminals to increasingly exploit the probable system weakness to conduct attacks and get access to sensitive patient information, or to impact the retrieved findings and device operations. On one hand, IoMT made patient life more sophisticated and adaptable. On the other hand, IoMT exposes users' privacy to increased threats/attacks. Furthermore, the IoMT security flaws are risky and may result in a life-threatening situation. Thus, the security of IoT devices has become a hot topic, and guaranteeing protection in the IoMT ecosystem is vitally crucial. Most health care experts are aware of the implications associated with IoMT security problems and this concern hampers the adoption of IoMT in medicine. During the last several years, the

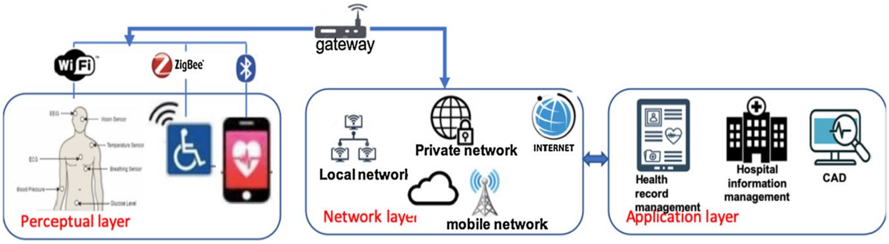


Fig. 1 General IoMT system architecture

IoMT system has been subjected to a number of diverse attacks prompting manufacturers and users to be more cautious while building and utilizing IoT devices [17]. Generally, the attacks specific to IoMT may be categorized into four: attacks on IoMT equipment, attacks on communication media, attacks on healthcare providers, and attacks on the patients. These cyberattacks may be launched in the context of IoMT by injecting malware into healthcare systems that prevents legitimate users from gaining access to certain areas of the system, for example, ransomware attacks. An intruder may also launch a Denial of Service (DoS) attack, which may result in hours, if not days, of disruption and unavailability of services.

3 Machine learning approaches for IDS

The main scope of our analytical study is to assess and compare the effectiveness of different ML algorithms for intrusion detection in IoMT networks. This study considers five ML algorithms and a brief recap of the five ML algorithms is presented as follows.

3.1 Naïve Bayes

A Naive Bayes is a simple and effective probabilistic ML algorithm. It is a special variant of the Bayesian network drawn from Bayes' theorem and the independence conditional naïve assumptions on individual features [18, 19]. This assumption ensures provides twofold benefits, first, it reduces the number of model parameters and estimates from a small number of training samples. Second, it improves the computational efficiency of NB. Although the NB assumption does not hold in real practice, it has surprisingly proven surprisingly effective in numerous domains, including on real-world data sets. The application of the Bayesian network classifier for intrusion detection can be expressed in mathematical form as follows [20]:

$$c(x) = \arg\max_{c \in C} P(c)P(a_1, a_2, \dots a_f|c) \tag{1}$$

In the equation above, the attack category c is determined if *f* features of the network traffic flow *x* is provided. Putting into play the "naïve" feature independence assumption, the above equation results in NB which classifies the given network traffic flow *x* as follows [19],

$$c(x) = \arg\max_{c \in C} P(c) \prod_{i=1}^{f} P(a_i|c) \tag{2}$$

Thus, the NB classifier recognizes intrusion detection using the posterior probability *P*(*ai*|c) and prior probability *P*(*c*) of network attack type *c*.

3.2 *K*-nearest neighbor (KNN)

KNN is one of the preferred ML algorithms for its conceptual simplicity for easy implementation. Yet, it is immensely powerful as it is based on nonparametric working principle which means it determines model structure with no assumption about the input network traffic data distribution. As KNN is an instance-based learning method, the principle idea employed for classifying new unknown attack traffic data involves two key steps [21], first resolves the nearest K training neighbors for the new attack traffic data based on distance/similarity measurement as shown in Fig. 2. Later, it classifies the new unknown attack traffic data using the majority votes of *k*-nearest neighbors. The most typically used similarity measurement is Euclidean distance which is described as the following [22],

$$D(x_1, x_2) = \sqrt{\sum_{i=1}^{f} \left(x_1^i - x_2^i\right)^2} \tag{3}$$

In the above equation, x_1 and x_2 are two instances with f features. Further, KNN being a variant of lazy learner, defers the training process until classification. As a result, it requires less training time than other classifiers. However, its testing phase computational expensive and slow, as processing of all training data takes place during testing phase. In the worst-case, KNN requires ample memory for storage and more time for classification process, when the training set holds large number of samples [12].

3.3 Decision tree (DT)

DT is a data driven ML algorithm that extracts and presents the knowledge in graphical tree structure with if–then-else decision rule for higher level of interpretability. It adopts greedy approach based on recursive partitioning for model construction

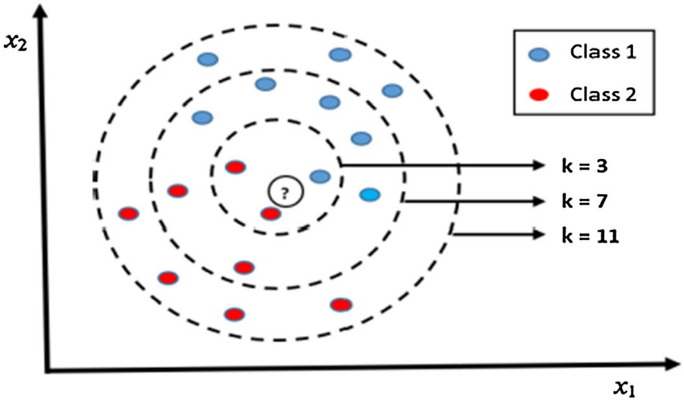


Fig. 2 Illustration KNN algorithm for binary classification

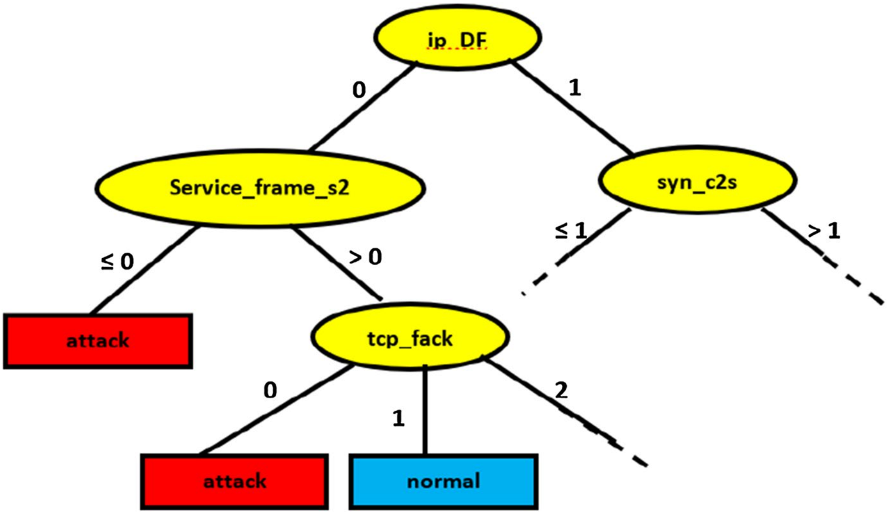


Fig. 3 DT with if–then–else rules for network traffic classification

[23]. The DT model contains two different types of nodes namely root/internal node with decision condition and terminal leaf node with class label as shown in Fig. 3.

It is constructed following two main steps, splitting and pruning [24]. The splitting process starts from root node with the given training set. At each decision node, it uses splitting measure to evaluate and find the feature with high discriminative power to split the available training set into subsets and assigned to the decision nodes at next level. The splitting process is repeated for all decision nodes until terminal leaf node is reached where all the available training instance belongs to same class. The pruning process aims to simplify the DT by removing unnecessary decision nodes and prevent data overfitting. This enables to improve the model generalization ability and reduce classification error. Thus, DT has several advantages as opposed to other ML algorithms such as presents better generalization ability with less number of model parameters, low computational load and memory requirement for model construction, robustness to noise and missing values, ability to handle redundant features.

3.4 Artificial neural network (ANN)

ANN is a special variant of ML algorithm that attempts to mimic the analytical behavior of human brain and enables to model complicated nonlinear relationships from the underlying data without making any prior assumptions about the data distribution [25]. ANN has many positive features over other ML algorithms. First, it has ability to learn fast and adapt its parameters for different kind of data. Second, it has stable generalization ability. Third, it has ability to fit data with arbitrary decision boundaries and improve the classification accuracy. Fourth, to improve the fault tolerance by reducing the sensitive to change in parameters. All these positive characteristics of ANN have made it gain popularity in the recent years as promising ML algorithm in various commercial and industrial applications.

Multilayer perceptron (MLP) is one of the most successful and practical ANN architectures with an input and an output layer but with a set of hidden layers [26]. The artificial neurons in the input layer receive input features xi from the given training set and pass to the hidden layer as shown in Fig. 4. The hidden layer also called the distillation layer, distills the important features and learns the complex relationships using the activation function which is defined as follows [27],

$$Y_i = Act\_fun\left(\sum_{N=1}^{f} W_{Nl}x_N\right) \tag{4}$$

Here, f and l denote the input features and number of neurons, respectively. Further, the activation function can be tanh, softmax, sigmoid, linear and rectified linear unit (RELU), and many others [28]. MLP network is trained using a backpropagation scheme to learn the optimal model parameters by adjusting the bias and weights at each epoch progressively against the obtained output error as given in Eq. (4). In this way, the scheme helps to achieve a gain in detection rate for malicious activities.

3.5 Support-vector machine (SVM)

SVM is one of the most popularly applied ML algorithms as a discriminative classifier leveraging the benefits of instance-based learning and convex optimization technique to construct a decision hyperplane for binary classification [13]. In this process, the instances that are very close to the hyperplane from both the classes called support vectors are determined. Then, the margin which is the distance from support vectors to the hyperplane is computed to find the optimal hyperplane with maximal margin. Thus, SVM employs the principle of structural risk minimization to find the global optimum hyperplane with

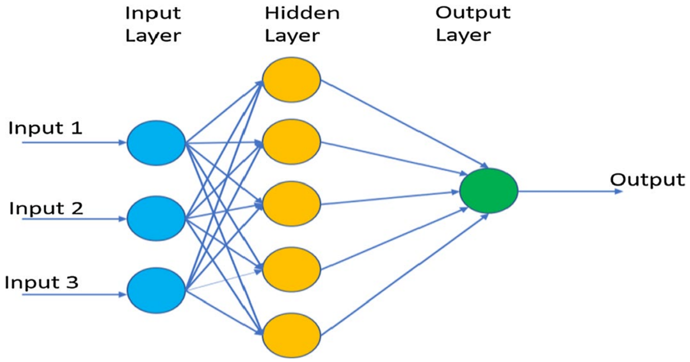


Fig. 4 ANN structure for network traffic classification

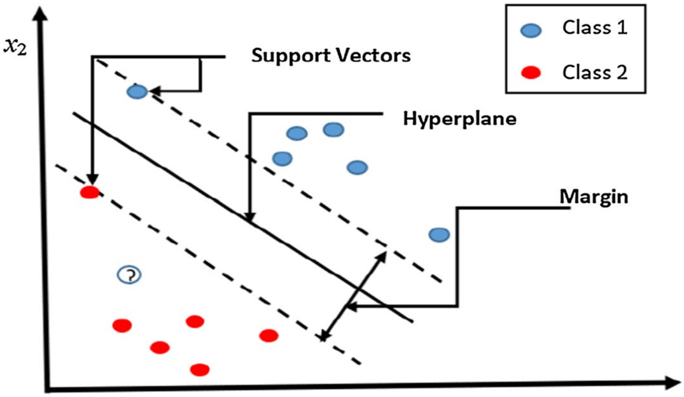


Fig. 5 Illustration of SVM algorithm for binary classification

maximal margin to avoid the risk of overfitting and achieve better generalization performance as shown in Fig. 5. The decision hyperplane is defined as [29],

$$\begin{aligned} &\vec{w}\cdot\vec{x}-b=0\\ &y_i\left(\vec{w}\cdot\underset{i}{\vec{x}}-b\right)\geq 1 \text{ for all } 1\leq i\leq n \end{aligned}\tag{5}$$

Here w and b are tuned with all n samples $\mathrm{x_i}$ during training process to meet the constraint given in Eq. (5). This enables to maximize the margin and correctly classify unknown new attacks.

In real practice, the intrusion datasets contain different attack classes that can be separable non-linearly in input space. In such scenarios, the original input is mapped to higher dimensional feature space applying kernels that support linear separation. The kernel functions most commonly used are radial, linear, and polynomial basis. The optimization problem is formulated as follows with kernel function for finding hyperplane [29],

$$\begin{aligned} &\min_{w,b,\xi}\left(\frac{1}{2}w^T w+C\sum_{i=1}^{l}\xi_i\right)\\ &\text{Subject to}\begin{cases} y_i\left(w^T\phi\left(x_i\right)+b\right)\geq 1-\xi_i\\ \xi_i\geq 0\end{cases} \end{aligned}\tag{6}$$

Here, ϕ represents kernel function, b represents hyperplane offset, w represents normal vector of the hyperplane, and C represents the penalty parameter of ξ the error term.

4 Experimental setup

In this section, the experimental setup followed to examine the performance of selected five ML algorithms for intrusion detection is described which includes datasets and evaluation framework.

4.1 Dataset description

Koroniotis et al. [30] presented the new dataset under the name Bot-IoT in 2018. It is one of the most recently published intrusion detection datasets for IoT environments and is publicly available for research purposes. The dataset was built at UNSW Canberra Cyber Range Lab simulating a realistic testbed environment for IoT Scenarios. The testbed for Bot-IoT dataset consists of VMs connected through LAN and Internet. The connection between VMs and the internet is established through PFSense system. Further, the IoT network with required IoT resources is simulated using an Ubuntu server. In the simulated IoT network, Kali Linux is utilized to launch attacks and ostinato utility is employed to produce normal network data traffic. Later, a realistic smart home network is developed utilizing five IoT devices that include remotely operational garage doors, a smart fridge, a weather station, motion-activated lights, and a smart thermostat. These devices are connected to cloud services using the node-red system to generate normal network traffic. Here, the IoT messages are transmitted to the cloud using MQTT protocol. Finally, Argus tool is employed to analyze the captured raw pcap files and extract 46 network traffic features. Figure 6 depicts the Bot-IoT attack taxonomy. There are four assault categories and eleven subcategories. An in-depth overview of the testbed settings and attacks is presented in [30].

The resultant original dataset contained 72 million records of legitimate and attack traffic flows. As recommended by the authors, the reduced dataset that represents 5% of the original dataset with the best ten features is used in this work

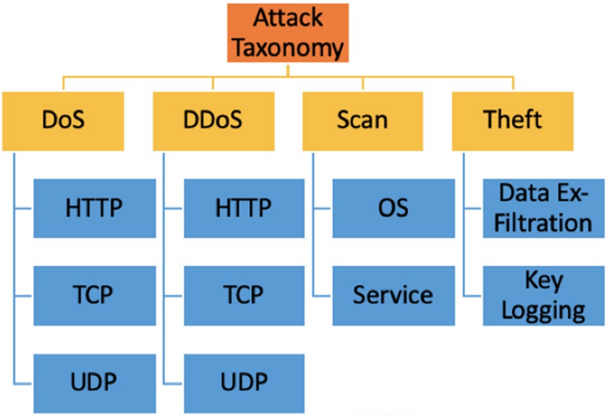


Fig. 6 Attack taxonomy of Bot-IoT Dataset

for all experimental evaluations. Table 1 illustrates the network traffic distribution across normal and attacks types. The observation of Table 1 clearly indicates that the dataset is class imbalanced to a greater extent.

This emphasizes the need for data sampling to balance all classes with enough instances and enable the ML algorithms to learn efficiently without being biased toward classes with more instances. To combat class imbalance, in this work we apply stratified random sampling. Table 2 illustrates the distribution of instance in the Bot-IoT training set before and after data sampling.

4.2 Experimental framework

This section presents the experimental framework designed to investigate the effectiveness of selected five ML algorithms for intrusion detection in IoMT network. The designed experimental framework shown in Fig. 7 consists of three phases, namely data preprocessing, hyperparameter optimization, and model evaluation. The main process carried out during these phases are briefed below.

Table 1 Data distribution for training and testing set of Bot-IoT dataset

IoT network traffic	Training set	Testing set
Attack	2,934,447	733,598
Normal	370	107

Table 2 Comparison of data distribution in the training set of Bot-IoT dataset before and after data sampling

IoT network traffic	Before sampling	After sampling
Attack	2,934,447	195,629
Normal	370	146,722

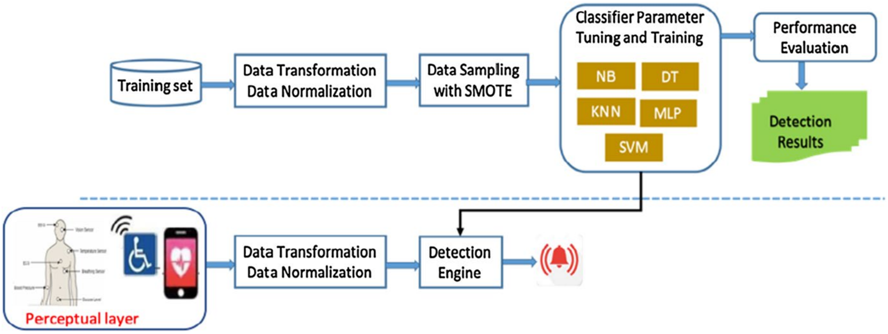


Fig. 7 Framework for investigating the effectiveness of ML-based IDS in IoMT environment

4.2.1 Data preprocessing

Data preprocessing is a process that plays an essential role in reducing the computational complexity of ML algorithms, transforming the raw data to useful format. This enables to enhance the efficiency of an IDS model for intrusion detection. Accordingly, the data preprocessing phases in this work involves three key operations, as follows.

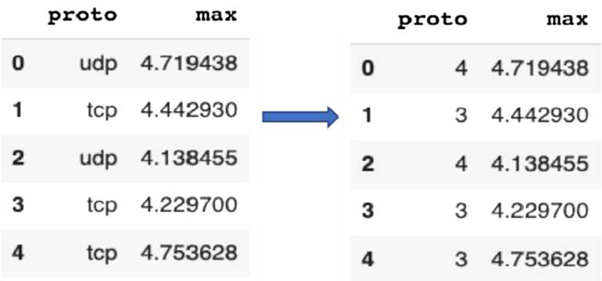
(a) *Data Transformation:* As machine learning models are not effective in handling string values, the label encoding method is used to map the symbolic features such as proto to numeric values [12]. The sample intermediate results from python environment that illustrates the label encoding for the feature proto in Bot-IoT dataset is shown in Fig. 8. Also, continuous features in the dataset Bot-IoT are discretized to enhance the performance of all ML algorithms that are chosen for investigation.

(b) *Data Normalization:* is a process of scaling the values of all features to the same scale. In doing so, all features can contribute proportionately and prevent the ML algorithms from being biased toward the features with larger values. Therein, considering normalization as an essential step in data preparation to enhance the prediction ability of the ML model, this work applies Z-score normalization [11] given in Eq. (7) to scale all features over the range [0,1] with mean (μ) and standard deviation (σ). The sample intermediate results from python environment that illustrates the data normalization for the feature max in Bot-IoT dataset is shown in Fig. 9.

$$x^* = \frac{x - \mu}{\sigma} \tag{7}$$

4.2.2 Hyperparameter optimization

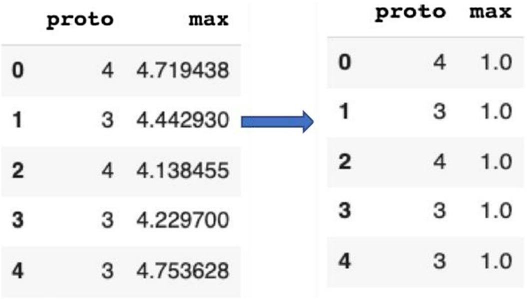
A hyperparameters are parameters whose value can govern the learning process. These parameters if tuned can boost the accuracy and generalization ability of



```
   proto       max
0    udp  4.719438
1    tcp  4.442930
2    udp  4.138455
3    tcp  4.229700
4    tcp  4.753628
```

```
   proto       max
0      4  4.719438
1      3  4.442930
2      4  4.138455
3      3  4.229700
4      3  4.753628
```

Fig. 8 Label encoding results of ‘proto’ feature in Bot-IoT dataset



	proto	max
0	4	4.719438
1	3	4.442930
2	4	4.138455
3	3	4.229700
4	3	4.753628

	proto	max
0	4	1.0
1	3	1.0
2	4	1.0
3	3	1.0
4	3	1.0

Fig. 9 Normalized results of ‘max feature in Bot-IoT dataset

the model. Hyperparameter optimization is very vital part of ML algorithms as it intends to optimize hyperparameters of ML algorithms and ensures to achieve maximum their performance during training process. To achieve fair comparison, the hyperparameters of all the five ML algorithms considered for investigation were carefully tuned employing grid search method with fivefold cross-validation. Unlike the existing literature on IoT environments that investigates the effectiveness of ML algorithms with default parameters for intrusion [15], our work explores to optimize parameters of all the chosen five ML algorithms to enhance detection accuracy while reducing the FAR. The parameter range utilized to initialize the grid search process and the results obtained are illustrated in Table 3. Further, the optimized hidden layer structure and model learning parameter of MLP is given in Table 4.

Table 3 Hyperparameter optimization for ML algorithms

ML algorithms	Parameters	Value range	Optimal value
NB	Var smoothing	{1e−5, 1e−10, 1e−20}	1e−20
KNN	*K*	{5, 10, 50, 100}	5
DT	Criterion	{‘gini’, ‘random’}	‘gini’
MLP	Activation function	{‘relu’, ‘logistic’,’tanh’}	‘relu’
SVM	Kernel	{‘rbf’, ‘linear’}	‘rbf’
	C	{100, 1000, 10,000, 100,000}	100,000
	γ	{10, 100, 1000, 10,000}	10,000

Table 4 MLP structure

Parameters	Values
Hidden layer structure	{25, 50, 75, 150}
Learning rate init	0.001
alpha	0.00001
Max iteration	100
Batch size	1024

4.2.3 Model evaluation

The fivefold cross-validation (CV) was applied as evaluation protocol to reduce the variation in results across data partitioning and to prevent model overfitting. In this process, the dataset is first shuffled randomly and then partitioned into five sets. Here, except one set others are used in training process. Thus, each of the designed experiments are executed five times on different data partition set for each ML algorithm chosen for investigation. Finally, the results obtained on the five sets are averaged to reduce variations and then reported for comparison. The averaged results from fivefold CV are compared and assessed computing the following metrics. The confusion matrix template defined by 2×2 matrix as shown in Fig. 10 is used to compute the following metrics.

- *Accuracy* is the fraction of correctly detected instances to the total instances in the testing set as given below

$$ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{8}$$

- *Precision* is the fraction of correctly detected attack instances to the total detected attack instances in the testing set as given below

$$PRE = \frac{TP}{(TP + FP)} \tag{9}$$

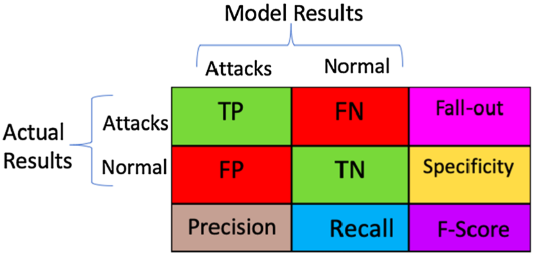
- *Recall* also called detection rate (DR) is the fraction of correctly detected attack instances to the total attack instances in the testing set as given below

$$DR = \frac{TP}{(TP + FN)} \tag{10}$$

- *F1-Score* enables to analyze the model performance combining both precision and recall metrics of a model and is computed as follows

$$F1 = \frac{2 \times (PRE \times DR)}{(PRE + DR)} \tag{11}$$

Fig. 10 Confusion matrix template for IDS evaluation



		Model Results		
		Attacks	Normal	
Actual Results	Attacks	TP	FN	Fall-out
	Normal	FP	TN	Specificity
		Precision	Recall	F-Score

- *False alarm rate (FAR)* is the fraction of normal traffic instances that are incorrectly detected to the total normal traffic instances in the testing set as given below

$$FAR = \frac{FP}{(TN + FP)} \tag{12}$$

4.3 Experimental setup

For all experiments in this paper, a Python notebook running on a web-based Google Colab portal with 12 GB of RAM and an HDD with more than 100 GB of storage space is used. Further, these experiments do not make use of a graphics processing unit (GPU). In our work, all chosen classifiers were built in Python using the Scikit-learn toolkit, which has a broad variety of cutting-edge ML methods. NumPy, SciPy, and matplotlib serve as the foundation for Scikit-learn. It is a simple and effective data mining and analysis tool. The Bot-IoT dataset was divided into sets: an 80% training set for optimizing models through a cross-validation process, and a 20% testing set for evaluating models and documenting testing results.

5 Results and discussion

The section concentrates to compare and assess the impact of five different ML algorithms chosen in this study for its effectiveness on both imbalanced (original) and balanced Bot-IoT dataset for intrusion detection performance from three aspects. First, the five different ML algorithms are analyzed for their intrusion detection ability with regard to three essential metrics namely ACC, DR, and FAR. Second, the chosen AE variants are examined to determine their stability for intrusion detection when trained with imbalanced datasets.

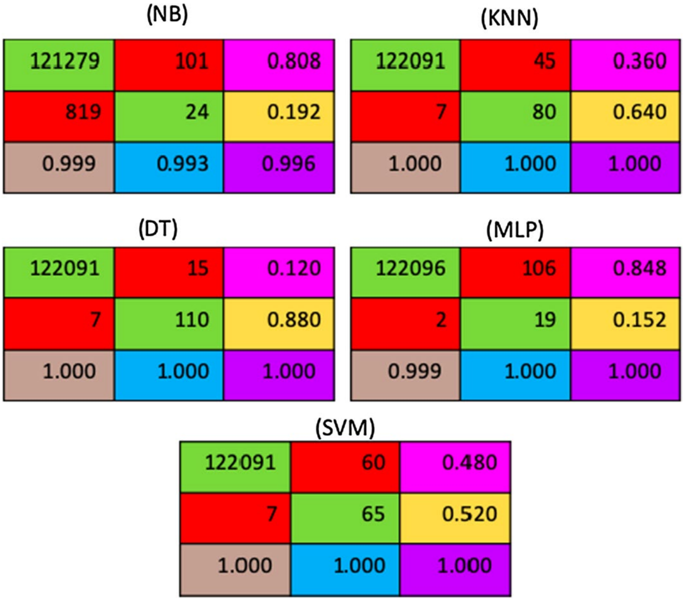
5.1 Performance analysis

The first step of analysis aims to investigate the application of the five different ML algorithms. In this regard, the detection performance metrics such as ACC, DR, and FAR discussed in the earlier section were computed for all the five ML algorithms and the results are reported in Table 5. To further improve the reader's understanding, additional performance metrics shown in Fig. 10 are also computed and presented in Fig. 11. The deep observation of the results remarks that DT achieves the best detection performance with all the three metrics, ACC, DR, and FAR on both imbalanced and balanced Bot-IoT datasets. Similarly, the second-best detection performance is demonstrated by KNN yet its performance is slightly degraded with regard to ACC and DR on the balanced Bot-IoT dataset but with improved FAR. Also, it can be seen that MLP and SVM gain performance improvement for intrusion detection on the balanced dataset with regard to FAR metric. But it can be noted that NB classifiers even with

Table 5 Comparative analysis of the chosen five different ML algorithms for intrusion detection on imbalanced Bot-IoT dataset

ML algorithms	Trained on imbalanced Bot-IoT			Trained on balanced Bot-IoT		
	ACC	DR	FAR	ACC	DR	FAR
NB	99.2	99.3	80.8	62.7	99.3	86.5
KNN	100	100	36	99.8	99.7	0.1
DT	**100**	**100**	**12**	**100**	**100**	**0**
MLP	99.9	100	84.8	99.3	99.6	0.9
SVM	99.9	100	48	99.8	99.6	0

The model with best performance measures are highlighted with bold fonts



(NB)

121279	101	0.808
819	24	0.192
0.999	0.993	0.996

(KNN)

122091	45	0.360
7	80	0.640
1.000	1.000	1.000

(DT)

122091	15	0.120
7	110	0.880
1.000	1.000	1.000

(MLP)

122096	106	0.848
2	19	0.152
0.999	1.000	1.000

(SVM)

122091	60	0.480
7	65	0.520
1.000	1.000	1.000

Fig. 11 Confusion matrix-based result analysis of the chosen five different ML algorithms for intrusion detection on a balanced Bot-IoT dataset

tuned hyperparameter fails to improve the detection accuracy in terms of ACC, DR, and FAR on both imbalanced and balanced Bot-IoT dataset.

On summary, it can be learnt that DT displays the best performance in terms of all the three key detection performance metrics disregarding the class imbalance effects. Further, to confirm that DT is the best choice among the five ML algorithms for building IDS model for intrusion detection in IoMT networks, the following section explores the ROC and PR analysis to visualize the impact of the best ML algorithms against intrusion detection in IoMT networks.

5.2 ROC analysis

This section uses the ROC curve, an acronym for receiver operating characteristic to compare the effectiveness of the chosen ML algorithms more intuitively for intrusion detection. In the literature, the ROC curve is considered one of the most essential metrics to assess the performance of ML algorithms for binary classifications on highly imbalanced intrusion detection datasets. This may be because it enables visualization of the model performance as a 2D graph plotting DR in relation to FAR; the two metrics which are regarded as a very crucial requirement for an IDS.

Accordingly, ROC curves for all the chosen five ML algorithms on imbalanced and balanced Bot-IoT datasets are illustrated in Fig. 12A, B, respectively. In an idle ROC curve, the binary classification with best performance approaches toward the upper-left corner. Based on this ground, the visual inspection of Fig. 12 clearly indicates that all the five ML algorithms display better performance on balanced dataset than on imbalanced dataset except NB algorithm. This confirms that all the five ML algorithms are sensitive to class imbalance. Certainly, this emphasizes the implication of data sampling in enhancing the performance of ML algorithms for intrusion detection. Further, to our surprise, observing the AUC values, it is evident that DT is effective and efficient compared to other ML algorithms to show intrusion detection performance of 94% and 100% under imbalanced and balanced situation, respectively. Also, it is appealing to note that DT displays DR of 100% and FAR of 0% on balanced dataset. This confirms the potential of DT for intrusion detection in IoMT networks. Thus, it is evident that DT can be recommended as the best ML algorithm for building IDS for IoMT networks and safeguard the network resources from sophisticated unseen cyberattacks.

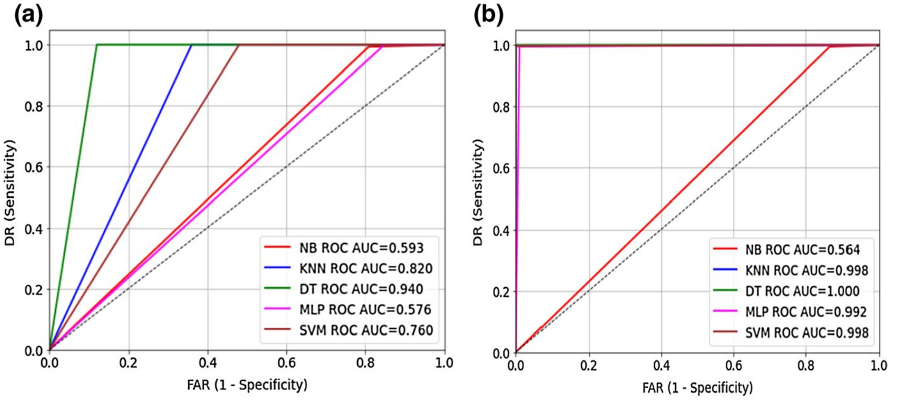


Fig. 12 Comparison of ROC curve for the five chosen ML algorithms on class imbalanced (**A**) and balanced (**B**) Bot-IoT dataset

6 Conclusion

This study investigates and presents a thorough comparison of five different ML algorithms for intrusion detection. As a first step, a recap on the working principles of the five chosen ML algorithms which includes NB, KNN, DT, ANN, and SVM is presented. Second, an experimental framework established to conduct a fair comparison is illustrated. Next, the hyperparameter of all the five ML algorithms are discussed with their optimal value to achieve the best intrusion detection performance. Finally, the detailed comparative results of all the chosen five ML algorithms are presented on both imbalanced and balanced Bot-IoT dataset. The analysis result demonstrates the superior performance of DT over other ML algorithms for intrusion detection. Further, one of the most essential metrics, ROC curve analysis is presented for all the chosen ML algorithms to confirm the effectiveness of DT over other ML algorithms for intrusion detection performance with relative importance to DR and FAR. The findings of this study sheds light to identify the best ML algorithm that can be employed to build effective IDS for IoMT networks. Therein, this study can be a starting point for researchers in the field of ML-based IDS to further explore and enhance the performance IDS for sophisticated unseen attacks. Indeed, in future study will concentrate to analyze and compare ML and deep learning algorithms for intrusion detection in IoMT networks.

Data availability Data sharing not applicable to this article as no datasets were generated during the current study.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Fontanet A, Autran B, Lina B, Kieny M, Karim S, Sridhar D (2021) SARS-CoV-2 variants and ending the COVID-19 pandemic. Lancet 397:952–954
2. Ravens-Sieberer U, Kaman A, Erhart M, Devine J, Schlack R et al (2021) Impact of the COVID-19 pandemic on quality of life and mental health in children and adolescents in Germany. Eur Child Adolesc Psychiatry 30:1–11
3. Siddiqui MF (2021) IoMT potential impact in COVID-19: combating a pandemic with innovation. Stud Comput Intell 923:349–361
4. Udgata SK, Suryadevara NK (2021) COVID-19, sensors, and internet of medical things (IoMT). In: SpringerBRiefs applied sciences and technology, vol 1, 1 edn. Springer, pp. 39–53
5. Joyia G, Liaqat R, Farooq A, Rehman S (2017) Internet of medical things (IoMT): applications, benefits and future challenges in healthcare domain. J Commun 4:240–247
6. Lu Y, Qi Y, Fu X (2019) A framework for intelligent analysis of digital cardiotocographic signals from IoMT-based foetal monitoring. Futur Gener Comput Syst 101:1130–1141
7. Vaiyapuri T, Binbusayyis A, Varadarajan V (2021) Security, privacy and trust in IoMT enabled smart healthcare system: a systematic review of current and future trends. Int J Adv Comput Sci Appl 12:731–737

8. Hameed S, Hassan W (2021) A systematic review of security and privacy issues in the internet of medical things; the role of machine learning approaches. PeerJ Comput Sci 7:e414
9. Alqaralleh BAY, Vaiyapuri T, Parvathy VS, Gupta D, Khanna A et al (2021) Blockchain-assisted secure image transmission and diagnosis model on Internet of Medical Things Environment. Pers Ubiquit Comput. https://doi.org/10.1007/s00779-021-01543-2
10. Khan S, Akhunzada A (2021) A hybrid DL-driven intelligent SDN-enabled malware detection framework for Internet of Medical Things (IoMT). Comput Commun 170:209–216
11. Binbusayyis A, Vaiyapuri T (2021) Identifying and benchmarking key features for cyber intrusion detection: an ensemble approach. IEEE Access 7:106495–106513
12. Binbusayyis A, Vaiyapuri T (2020) Comprehensive analysis and recommendation of feature evaluation measures for intrusion detection. Heliyon 6:e04262
13. Binbusayyis A, Vaiyapuri T (2021) Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM. Appl Intell 51:7094–7108
14. Vaiyapuri T, Sbai Z, Alaskar H, Alaseem NA (2021) Deep learning approaches for intrusion detection in IIoT networks—opportunities and future directions. Int J Adv Comput Sci Appl 12(4):86–92
15. Shafiq M, Tian Z, Sun Y, Du X, Guizani M (2020) Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. Futur Gener Comput Syst 107:433–442
16. Sun L, Jiang X, Ren H, Guo Y (2020) Edge-cloud computing and artificial intelligence in internet of medical things: architecture, technology and application. IEEE Access 8:101079–101092
17. Kumar P, Gupta GP, Tripathi R (2021) An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. Comput Commun 166:110–124
18. Lewis DD (1998) Naive(Bayes)at forty: the independence assumption in information retrieval. Lect Notes Comput Sci 1398:4–15
19. Ziegel ER (2001) Bayes and empirical Bayes methods for data analysis. Technometrics 43:246–255
20. Kass RE, Raftery AE (1995) Bayes factors. J Am Stat Assoc 90:773–795
21. Anava O, Levy KY (2016) k*-Nearest neighbors: from global to local. In: Proc. NeurIPS, Barcelona, Spain, pp 4923–4931
22. García-Pedrajas N, Castillo JD, Cerruela-García G (2015) A proposal for local k values for k-nearest neighbor rule. IEEE Trans Neural Netw Learn Syst 28:470–475
23. Swain P, Hauska H (1977) The decision tree classifier: design and potential. IEEE Trans Geosci Electron 15:142–147
24. Ferrag MA, Maglaras L, Ahmim A, Derdour M, Janicke H (2020) Rdtids: Rules and decision tree-based intrusion detection system for internet-of-things networks. Future internet 12:44–57
25. Hopfield J, Tank D (1986) Computing with neural circuits: a model. Science 233:625–633
26. Yu L, Wang S, Lai K (2007) Basic learning principles of artificial neural networks. In: Foreign-exchange-rate forecasting with artificial neural networks, vol 107. Springer, pp 27–37 (2007)
27. Mehrotra K, Mohan C, Ranka S (1997) Elements of artificial neural networks. MIT Press
28. Vaiyapuri T, Binbusayyis A (2021) Application of deep autoencoder as an one-class classifier for unsupervised network intrusion detection: a comparative evaluation. PeerJ Comput Sci 6:e327
29. Vaiyapuri T, Binbusayyis A (2021) Enhanced deep autoencoder based feature representation learning for intelligent intrusion detection system. Comput Mater Contin 68(3):3271–3288
30. Koroniotis N, Moustafa N, Sitnikova E (2019) Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Futur Gener Comput Syst 100:779–796

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Adel Binbusayyis[1] · **Haya Alaskar**[1] · **Thavavel Vaiyapuri**[1] · **M. Dinesh**[2]

✉ Thavavel Vaiyapuri
t.thangam@psau.edu.sa

[1] College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University,
Al Kharj, Saudi Arabia

[2] College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia