Wireless Networks (2021) 27:2761–2784
https://doi.org/10.1007/s11276-021-02594-2

Computational intelligence techniques for automatic detection of Wi-Fi attacks in wireless IoT networks

M. Nivaashini[1] · P. Thangaraj[1]

Accepted: 2 March 2021 / Published online: 15 April 2021

Abstract
These days, number of smart products based on Internet-of-Things (IoT) has been increased. These products are unified via various wireless technologies like, Bluetooth, Z-wave, Wi-Fi, Zigbee, etc. While the need on the wireless networks has improved, the assaults against them throughout the time have expanded on top. In order to identify these assaults, an intrusion detection system (IDS) with a prominent precision and low identification time is required. In this work, a machine learning (ML) based wireless intrusion detection system (WIDS) for wireless networks to effectively identify assaults against them has been proposed. A ML prototype has been implemented to categorize the wireless network records into ordinary or one of the particular assault categories. The operation of an IDS is extensively enhanced when the attributes are more discriminative and delegate. Different attribute selection methods have been investigated to identify the best set of attributes for the WIDS. The proposed model is evaluated on aegean wireless intrusion dataset using various parameters like attack detection rate, detection time, precision, F-measure, etc. The experimental evaluation is carried out in the tools like, Weka, Rstudio and Anaconda Navigator Python. Finally, the experimental result shows the best performing ML algorithm with best set of reduced attributes.

Keywords Intrusion detection system · Wireless attacks · Machine learning · Attribute reduction · Attack classification

1 Introduction

The new advancement in telecommunications and data innovations, for example, the Internet of Things (IoT), has exceptionally outperformed the conventional detecting of neighboring circumstances. IoT innovations have encouraged the advancement of frameworks that can improve life quality. IoT is one of the quickest developing advances in computing, with an expected 50 billion gadgets before the finish of 2020 [1]. It has been assessed that, by 2025, the IoT and associated usages have a possible financial effect of $3.9 trillion to $11.1 trillion every year [2].

The wireless network and IoT are all inclusive growing, giving assorted advantages in almost each part of our lives [3–7]. Tragically, the IoT is likewise joined by countless data security vulnerabilities and endeavors [3, 5, 7–12]. Wireless network data are relied upon to increment quickly because the wireless network is a typical system for minuscule gadgets spread anyplace as IoT become progressively well-known nowadays [13]. Inappropriately, weaknesses and attacks for IoT devices in wireless systems are developing subsequently [13]. Moreover, since IoT gadgets usually function in an unattended circumstance, an attacker might genuinely get to these gadgets with malevolent intention [14]. Additionally, on the grounds that IoT gadgets are associated normally over wireless networks, snooping can be utilized to get into the personal data from a correspondence station [15]. Therefore, IoT frameworks are more defenseless when contrasted with conventional processing frameworks. This requires study in explicit detective and preventive methods for IoT frameworks to ensure protection against wireless network attacks.

Information produced by the wireless networks of IoT gadgets is enormous and subsequently, conventional

✉ M. Nivaashini
nive19794@gmail.com

P. Thangaraj
hod_cse@kpriet.ac.in

[1] KPR Institute of Engineering and Technology, Coimbatore, India

information collection, storing, and handling strategies may not function at this measure. Besides, the sheer measure of information can likewise be utilized for designs, practices, forecasts, and evaluation. Furthermore, the divergency of the information produced by IoT makes additional front for the current information preparing procedures. Consequently, to tackle the estimation of the IoT-produced information, new components are required. In this framework, ML is observed as the most appropriate computational standards to give fixed insight in the IoT gadgets [16]. ML can support technologies and smart gadgets to induce valuable information from the gadget or human-created information. It can be characterized as the capacity of a smart gadget to fluctuate or mechanize the circumstance or practices on information which is measured as a fundamental part for an IoT plan. ML strategies have been utilized in undertakings, like, categorization, regression and density assessment. Collection of utilizations, like, computer vision, fraud detection, bio-informatics, malware detection, authentication, and speech recognition utilize ML methods and procedures. Similarly, ML can be utilized in IoT for offering intellectual facilities. In this paper, on the other hand, the uses of ML in giving security and protection facilities to the wireless networks of IoT.

ML is the investigation of techniques that progress their functioning with knowledge and are intended to mechanize practices; the machine makes each vital stage perfectly in a sustained manner. It is a kind of artificial intelligence (AI) that furnishes personal computers (PCs) with the capacity to study mechanically without being unambiguously programmed. ML indicates intellectual techniques used to enhance the execution standards utilizing model information or former knowledges via learning. More exactly, ML techniques construct models of practices utilizing numerical methods on enormous informational collections. These models are utilized as a foundation for making upcoming forecasts built on a fresh input information. ML is used when individual skill either don't endure or can't be utilized, like, exploring a threatening spot where people can't utilize their skill, for example mechanical technology, speech recognition and so on. It is likewise employed in circumstances where answer for some particular issue changes as expected (steering in a PC organization or finding malevolent code in a product or application).

Moreover, it is utilized in functional brilliant frameworks, for example Google utilizes ML to investigate dangers against mobile terminations and appliances operating on Android. It is likewise utilized for recognizing and eliminating malware from tainted phones. It incorporates different learning strategies categorized as supervised, unsupervised and reinforcement learning relying upon the occurrence or the absence of named information. Supervised learning prepares the program with named trials; thus, the prepared program can anticipate alike unnamed trials. It incorporates prediction, knowledge mining and compression actions. Unsupervised learning doesn't have any preparation trials; it utilizes the factual methodology of density assessment. It performs by the rule of finding the concealed plan of the information by bunching or gathering information of comparable type. It incorporates mechanisms like pattern recognition and outlier recognition. Reinforcement learning is determined on programming specialists that want to make a move in a circumstance so it amplifies total prize [17]. Each progression of the specialist isn't studied completely for progress or disappointment yet on a grouping of activities taken together ought to have a way in the direction of great strategy. This learning is tremendously utilized in Gaming hypothesis and Robot Navigation.

Though conventional methodologies are generally utilized for various parts of IoT (for example applications, facilities, designs, conventions, information accumulation, source distribution, grouping, examination) comprising security, the enormous scope utilization of IoT. Though, they do not have the capacity to mechanically identify new assaults. Since network conditions change rapidly, assault variations and new assaults arise continually. Subsequently, it is important to create IDSs that can recognize new assaults mechanically. To tackle the above issues, scientists have started to concentrate on developing IDSs utilizing ML methods. ML is a kind of computationally intellectual methods that can consequently find helpful data from huge datasets. ML-based IDSs can accomplish good discovery levels when adequate training information is free, and ML models have adequate generalizability to distinguish assault variations and new assaults. Subsequently, ML-based IDSs don't depend vigorously on field information; so, they are simple to plan and develop. Finally, ML are confirming methods for IoT networks because of numerous causes, for example IoT networks generate complete measure of information which is needed by ML techniques to get intellect to the frameworks. Moreover, the value of the information created by the IoT is better used with the ML strategies which empower the IoT frameworks to build on knowledgeable and intellectual choices. ML procedures are generally utilized for security, protection, assault identification and malware examination. An IDS utilized with different ML methods is a phenomenal identifier of wireless network attacks in IoT.

Inside the more extensive zone of system security examine, there are many research exercises that mean to improve interruption identification methods. While continuous interruption identification is a significant component of an IDS, maximum number of IDSs work in disconnected approach because of the necessity to

investigate an enormous measure of system movement information. Meant for investigation, disconnected approach gives chance to top to bottom investigation of examples and practices of intrusions. Moreover, it gives chances of testing for interruption identification methods. In working perspective disconnected procedure gives top to bottom examination of previous information and produce avoidance techniques for upcoming events. By means of checking huge quantity of attributes for attack identification may increment time complexity of the IDS model. An IDS routinely handles massive amounts of data traffic that contain redundant and irrelevant attributes, which impact the performance of the IDS negatively. Attribute selection methods play an important role in eliminating unrelated and redundant attributes in IDS. Statistical analysis, neural networks, ML, data mining techniques, and support vector machine models are employed in some such methods. Good attribute selection leads to better classification accuracy. So, it is essential to choose the superlative attributes that proficiently support attacks identification procedure.

Figure 1 shows the major design of the IoT. IoT design has three layers specifically, the perception, network, and application layers. (1) The perception layer is the actual layer, which has sensors for detecting and assembling data about the circumstance. It detects some actual limits or recognizes other smart gadgets in the circumstance. (2) The network layer is liable for associating with other brilliant things, network gadgets, and servers. Its features are likewise utilized for sending and preparing sensor data. (3) The application layer is answerable for conveying application explicit facilities to the client. It describes different applications in which the IoT can be implemented, for instance, smart homes, brilliant urban communities, and smart health. Every layer in the IoT design has hold its individual set of security dangers. Although, Network layer turns like a bond between perception and application layer. It conveys and sends the data gathered from the actual things through sensors. The mode for the broadcast can be wireless or wired. It additionally assumes the liability for interfacing the smart objects, network gadgets and networks to one another. Thus, it is greatly delicate to assaults from the side of attackers. It has unique security problems

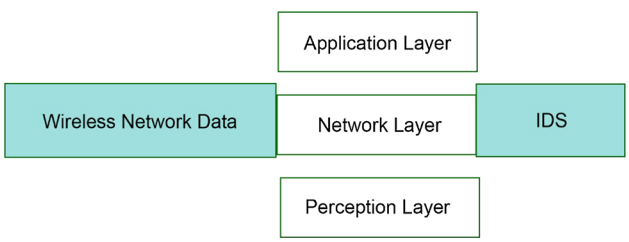

Fig. 1 IoT Architecture with IDS

with respect to integrity and authentication of data that is being moved in the network. Subsequently, the proposed work essentially concentrates around the usage of WIDS in the network layer of the IoT design.

1.1 Attacks in the layers of IoT architecture

1.1.1 Perception layer attacks

In perception layer assaults, the attackers have straight permission to the gadgets and control various parts of the gadgets. To gain permission to the actual gadgets, social designing perhaps the most obvious strategies where the attackers contact the gadgets and accomplish genuine assault that goes from actual harm to the gadget to snooping, side-channels, and other related assaults [8, 18].

1.1.2 Network layer attacks

At the network level, the assaults are pointed towards channeling, information and traffic investigation, spoofing, and dispatching man-in-the-middle assault. Furthermore, sybil assaults are additionally conceivable at the network layer where counterfeit characters/sybil personalities are utilized to make impressions in the network [19, 20]. With these assaults, the chance of dispatching a shared Distributed Denial of Service (DDoS) assaults increments, and subsequently upsetting the entire IoT network. At the network layer, the attacker can accomplish this by bombarding the organization with more traffic through conceded hubs than the network can deal with [21].

1.1.3 Application layer attacks

IoT applications are generally worthwhile focuses for the attackers since applications level assaults are moderately simple to dispatch. A portion of the notable assaults incorporate, yet not restricted to, buffer overflow assaults, malware assaults, DoS, phishing, misusing the WebApp weaknesses, cryptographic assaults, side channel assaults, and man-in-the-middle assaults. Buffer overflows are one of the generally utilized assault vectors in various applications [22]. IoT applications are additionally inclined to malevolent code infusion because of buffer overflow and different weaknesses, for example, SQL infusion, cross-site scripting, object referring, etc.

1.2 Principle of the study

Proof demonstrates that a large portion of the assaults are created from the network layer [23]. The point of this examination is to build up an answer, assembling an intellectual IDS that can identify the attackers and block

assaults in the wireless networks of IoT. To accomplish this objective, initially, the present status of ML methods-based IDS models is inspected. To recognize expected security and protection problems associated with the wireless network assaults which have previously happened in the IoT frameworks are analyzed. At last, an answer is proposed, using the strength of ML methods to battle unexpected assaults from unapproved attackers, in the wireless network of IoT as appeared in Fig. 1. To advance the exhibition of IDS, different feature enhancement procedures are investigated.

The proposed system can answer the following problems:

1. What attributes best represent different attacks?
2. What type of data is most suitable for detecting certain attacks?
3. What types of ML algorithms are the best fit for a specific data type?
4. How do ML methods improve IDSs along different aspects?

Ever since, various investigations have utilized more established data collections, like, NSLKDD [24], KDDCUP 99 [25] and numerous analysts show that these data collections are obsolete nowadays [26, 27]. Thus, it is imperative to assess novel data collections in order to supplant these ancient set of data collections.

This study has assessed the openly accessible Aegean Wireless Intrusion Dataset (AWID) [17] through various ML and attribute reduction methods in the wireless network of the IoT network layer architecture as shown in Fig. 1. The AWID data collection comprises of actual hints of both legitimate and benign data packets of 802.11 Wireless network system and can be categorized into two parts namely, high-level marked data collection along with four significant type of data packets and finer grained data collection. Using ML and attribute reduction methods, the data packets in the AWID data collection can be detected either as a legitimate or a particular interruption category. The AWID datasets can be mainly classified into two types based on class labeling, the high-level labelled dataset contains 4 major classes while other dataset has a finer grained class labelling. Five traditional attribute reduction techniques i.e., Information Gain (IG), Correlation-based Feature Selection (CFS), Chi-Squared statistics (CH), Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) are tested in this examination. The appropriate execution of the attribute reduction techniques is examined on AWID utilizing six ML algorithms, i.e., Support Vector Machine (SVM), Naïve Bayes (NB), J48, Multi-Layer Perceptron (MLP), Random Forest (RF), and k-Nearest Neighbor (kNN).

The rest of the paper is organized as follows: related work is presented in Sect. 2. The fundamental working process of the different attribute reduction techniques and ML algorithms are described in Sect. 3 and working methodology of the proposed model is presented in Sect. 4. Experiments conducted and results obtained are elaborated in Sect. 5. Lastly, conclusion and future scope of the research are briefed in Sect. 6.

2 Related work

In this section, former methods utilized in the identification and categorization of different Wireless attacks and network attacks which happens in the wireless network utilizing various attribute reduction techniques and ML classifiers have been presented along with their limitations.

2.1 Attribute reduction techniques used in AWID wireless dataset

The works in [28–30] have used neural networks and clustering techniques in the implementation of the IDS with the typical MAC header attributes in the learning process. Because of using only, the MAC header attributes the computational complexity of time and capacity have been increased in the former works, which in turn diminishes the efficiency of the IDS. Therefore, it is evident from the previous work that, the insignificant and repetitive attributes might produce an interference in the training procedure and corrupt the attack recognition precision of the IDS. In order to overcome this the authors of the work [31–33] have used artificial immune system-based attribute selection techniques for selecting an optimal set of attributes, which in turn improves the efficiency of the IDS. An optimum set that has a more prominent interruption discovery inclination is chosen to develop the appropriate identification methods. The proposed work in [31–33] have failed to address the problem of processing cost in the attribute selection process. Therefore, there rises a need to choose an optimum attribute selection method in the implementation of the IDS. In [34] authors choose Mutual information-based attribute selection method in order to implement an efficient IDS with higher performance. The crucial issue in [34] is to pick an optimum set of attributes pertinent to the learning calculation. To handle the issue of attribute reduction, a Novel Hybrid Mechanism (NHM) has been proposed in [35] that chooses the optimum attributes for defining intrusions in 802.11 systems and joins the wrapper and filter methods to choose ideal attributes. The filter method figures the data gain for every attribute, and positions the attribute using the calculated gain score. It chooses just the top positioned attributes utilizing a limit

worth, and it utilizes k-means grouping algorithm to choose an optimum list of top positioned attributes. The optimum list of attributes determination may decrease the incorrect identifications. In [35] the k-means grouping algorithm neglects to quantify the quantity of groups precisely once the data collection is vast.

Other exertion in [36] moreover uses an amalgam strategy that expels repetitive attributes utilizing weighted mutual information. But an amalgam strategy discards the attribute repetition with the ML algorithms because of costly algorithm. Because of what, it can't decide all the potential amalgamations of attributes. Thus, it leads to an opportunity of losing some amalgamation that seriously influences on the precision of IDS. The work in [37] used an amalgam strategy coordinating Genetic algorithm and Latent Dirichlet Allocation, termed as G-LDA chooses an optimum subgroup of attributes based on average and high frequency esteems to distinguish the malicious packet data. But, G-LDA neglects to study the various count of class samples in pre-processing step and therefore, it propels to conceal the significance of the maximum important attributes.

A trivial IDS chooses an optimum attribute subgroup utilizing wrapper method in [38], and it gives the list of attributes to neuro tree algorithm to accomplish great caliber execution in neural systems. Still, it might not be appropriate to deal with huge estimated data collection progressively. To scan optimum set of attributes for these kinds of huge datasets, in [39] Genetic Algorithm (GA) has been utilized with the intend to expand the identification exactness and to diminish the incorrect identifications in interruption identification. Still, this strategy is additionally tedious because of its powerfully coupled induction method with the recurrently working technique to assess the presentation of every subgroup of attributes. An imperative list of attributes is chosen by the consecutive reverse procedure in [40]. It expels single attribute in every cycle, and a few such repetitions are completed by choosing an optimum list of attributes till the algorithm reaches a specific limit. But it consumes more processing time as it eliminates single attribute per iteration. In [41] a heuristics-based feature selection method called CFS-BA techniques has been used for the selection of optimal set of features and combines C4.5, RF, and Forest by Penalizing Attributes (Forest PA) algorithms. But its capability could be further improved to deal with rare attacks from the massive network traffic. Cross—Correlation based Feature Selection (CCFS) method is implemented in [42] using four different classifiers: SVM, NB, decision tree, and kNN. The proposed method uses CCFS for an optimal set of feature selection. But it neglects to study other feature selection methods.

2.2 ML techniques used in AWID wireless dataset

Notwithstanding the affirmation of the striking execution of IDS utilizing attribute reduction, there are a few integral difficulties in the identification and grouping of attacks for incorrect identification and precision measurements. Different ML algorithms namely, NB, kNN, neural systems, decision trees, rule-learners, and SVM are utilized in [43]. Additionally, it analyzes the exhibition of grouping algorithms and examined those algorithms. Seven types of classifiers are investigated in [44] and it concludes that the Decision Trees, kNN, SVM and C4.5 provide high performance than others. But the classifiers are used on the simulation data and the generated datasets that are not dependent to a special problem. An innovative operator-based IDS has been proposed in [45] that utilizes rough set theory to group attacks, and to accomplish noise and vulnerability in information. Yet, the rough set algorithm increases the execution cost, and it needs the information on complete set of attributes to categorize intrusions. In [46] NB utilizes a Bayesian method for categorizing the attack occurrences in the system and it is implemented using separate individual hypothesis between attributes that fundamentally expands the exactness of IDS. Notwithstanding these favorable circumstances, the capacity and computing time complications of NB dependent IDS are one-sided for the massive data collection.

An IDS dependent on k-means grouping and Particle Swarm Optimization (PSO) algorithm has been proposed in [47], that decreases the neighborhood minima by means of its utilization of PSO for distinguishing the wellness of the information. This technique viably identifies new attacks on the system; however, it builds the incorrect identification percentage once the quantity of new attack increments. A supportive system IDS [48] uses the fuzzy based SVM to recognize attacks associated with various system conventions. It partitions the system information with a system convention and progresses the rapidity of recognition specialist and the precision percentage of identification. Still, it decreases the identification precision, once there is an unexpected attack. The qualities of the SVM algorithm are utilized to classify the attack design from the ordinary one using pre-established group of past data [49]. Still, other data mining techniques such as genetic algorithms, case-based reasoning, decision tree and inductive learning may be applied to IDSs. Because comparisons of various data mining techniques will provide clues for selecting appropriate models for detecting intrusions. An IDS using single label SVM has been proposed in [50], that remarkably alters the attack and ordinary sketches and accomplishes a superior identification precision in any event, once the samples of training data are minimum and

maximum. The implementation in [49, 50] separates individually the attack and the best outline. Meanwhile it neglects to group the irregular attack design beneath an alternate sort of attack class types. To turn away this, [51] implements a multi-class SVM that effectively separates the unstable attack beneath particular class labels, however the precision of IDS has been least, because the multi-class SVM needs the data of entire set of training data collection.

A combined approach PSO–SVM has been anticipated for an interruption identification issue, the PSO is utilized to decide permitted limits of SVM algorithm and to get an ideal subset of attributes by implementing IDS [52]. This technique fails to address the measurement of attribute reduction process. This method neglects to report the estimation of attribute reduction procedure. A novel IDS has been implemented in [53] that uses the PSO to choose the finest subset of attributes and to select SVM limits and deploys IG attribute reduction method for attribute reduction process. The SVM–PSO gains maximum recognition exactness than an ordinary SVM algorithm. But the IG is one-sided, as soon as the attributes have extra discrete qualities, and it diminishes the value of the attribute subset selection. In [54] an IDS is implemented using SVM on top of RBF to achieve maximum identification precision and chooses uppermost attributes to diminish the computational complexity of the classification process. However, it fails to achieve higher accuracy with other kernel functions of SVM like linear, Gaussian and Polynomial. In [55] a wrapper method is utilized for attribute reduction and limit enhancement in SVM. The consistent PSO is used to improve the limits of the RBF based SVM, and the twofold PSO is utilized to choose ideal attributes, bringing about maximum precision percentage in attack identification process. Yet, only single data collection with a few attributes is utilized in this examination, which doesn't show the complete execution of the anticipated algorithm.

The AWID high-level marked data collection were initially examined with various ML classifiers in [56]. In [57], AWID data collection is tested with seven notable ML algorithms, i.e., RF, AdaBoost, ZeroR, J48, Random Tree, MLP, and logit Boost accompanied by CFS evaluator in place of attribute reduction method. Still, it can use other attribute reduction techniques than CFS evaluator for identifying better optimal set of attributes in detecting various attacks in the WIDS.

3 Background

3.1 Attribute reduction methods

Attribute reduction method is the greatest significant piece of work that will improve the efficiency of the categorization model, as the attribute reduction techniques will choose the maximum investigative attributes. Attribute reduction techniques decrease the unique list of attributes via eliminating insignificant attributes aimed at wireless attack identification to progress categorization precision and decrement the recurring period of learning algorithms. The proposed study has examined the efficiency of the five generally utilized attribute reduction approaches in the ML investigation, namely, IG, CFS, CH, PCA, & LDA are briefly explained in the following sub-sections. Wholly these attribute reduction approaches are utilized to process a rank value meant for every single attribute and at that point a pre-established count of attributes are chosen according to positions acquired from the rank value. Also, the determination of the attribute reduction techniques is affected by the information capacity, information constancy, and the necessity to explore the most effective attribute reduction approach is required.

3.1.1 Information gain (IG)

The IG technique is utilized to choose the attributes with maximum analytical information, which helps in upgrading the categorization process of the data from the unique information index. IG method assesses the value of an attribute through estimating the IG value as per the class label. IG depends on the idea of entropy which is broadly utilized in the data hypothesis area. Assumed a group of samples S, comprising legitimate and malicious samples of some objective idea. The entropy of S, comparative with this Boolean grouping is specified via:

$$Entropy(s) = Info\ Gain(IG) = \sum_{i}^{m} -Plog_2(P) \quad (1)$$

$$Gain(S, A) = Entropy(s) - \sum_{v \in Values(A)} \frac{|s_v|}{s} Entropy(s_v) \quad (2)$$

Here IG is determined by computing the likelihood of event of class label over complete class labels in data collection, where Pi is the arbitrary likelihood that a subjective instance has its place in the class label Ci [58, 59]. A is the collection of every single esteem of attribute A and S_v is the subgroup of S intended for the attribute A of esteem v. On the whole, only the highly ranked attributes are used by the ML classifiers to categorize the collected data collection as either legitimate or malicious.

3.1.2 Correlation-based feature selection (CFS)

CFS surveys the estimation of the collection of the attributes by means of reviewing the specific investigative capacity of every attribute along with the likelihood of

recurrence between the attributes. CFS assesses the attributes which are exceptionally connected with the class, still disconnected with one another [60].

$$R = \frac{\sum_1^N (a_\mathrm{i} - A')(b_\mathrm{i} - B')}{N\sigma A \sigma B} \tag{3}$$

$$R = \frac{\sum_1^N (a_\mathrm{i} b_\mathrm{i}) - NA'B'}{N\sigma A \sigma B} \tag{4}$$

where N is the count of columns i and i is the individual estimations of A and B in the ith column, A and B are the separate average estimations of A and B, σA & σB are the particular normal deviations of A and B. The valuation of R arrays amid − 1 and 1.

3.1.3 Chi-squared statistics (CH)

The CH attribute reduction signifies the relationship amongst the attributes and the comparing categorical output. The disparity from the normal dispersion is estimated through the measurable assessment dependent on presumption that the attribute existence is autonomous of the last categorical output [52, 53]. It is characterized as,

$$CHI(t, c_\mathrm{i}) = \frac{N \times (AD - BE)^2}{(A + E) \times (B + D) \times (A + B) \times (E + D)} \tag{5}$$

$$CHI_{max}(t) = max_\mathrm{i} CHI(t, c_\mathrm{i}) \tag{6}$$

Here A is the rate of recurrence as soon as t and i happens together;B signifies the count of occurrences once t happens in the absence of c_i. E signifies count of occurrences once c_i happens in the absence of t; D is the rate of recurrence once neither c_i nor t happens and N is the entire list of samples in the report collection. The CH measurement will be nil uncertainty t and c_i are free.

3.1.4 Principal component analysis (PCA)

The maximum utilized and greatest well-known attribute reduction procedure is PCA [61], which is existing because of its estimation workability and adjustable methodology. PCA is accomplished by the removal of least important attributes from top to bottom proportional area that establishes the key computing expense and anticipating the greatest supportive valid attributes obsessed by a shallow proportional subarea resulting in a less difficultly. At the end, on the condition that the cross product is of dimension 'm' and adjacent are 'n' perceptions then the lattice (A) can be denoted by means of.

Each segment is a cross product, subsequently appeared as in Eq. (7).

$$A_{m\times n} = [A_1, A_2, A_3 \ldots, A_n] \tag{7}$$

The normal average value () of each cross product is determined utilizing Eq. (8).

$$Mean,\ \mu = \frac{1}{n}\sum_{i=1}^{n} A_{ij} \tag{8}$$

Variance is a measurement utilized to determine the variety of a cross product forms from its normal average value. The variance is computed by means of Eq. (9).

$$\Phi = A_i - \mu \tag{9}$$

The co difference is a measurement utilized to quantify the level of connection among the dual factors. The optimistic estimation of the outcome demonstrates the dual factors are optimistically associated, whereas the adverse estimation looks like the adversely associated information and the nil estimation proposes that the information isn't associated. The growth of the information is clear via the co divergence lattice. The co divergence lattice is latter built over the square lattice accompanied by the no. of classes as the measurement as shown in Eq. (10).

$$B_{m\times n} = \frac{1}{n-1}\sum_{i=1}^{n} \Phi^i \Phi_i^t \frac{1}{n-1}\sum_{i=1}^{n} (X_i - \mu)(X_i - \mu)^t \tag{10}$$

where Φ^t is the invert of the lattice Φ.

To achieve PCA over the variance lattice, computation of Eigen-qualities and Eigen cross products are commonly utilized via Singular Value Decomposition (SVD). State $(\lambda_1, \mathrm{u}_1)$, $(\lambda_2, \mathrm{u}_2)$ …, $(\lambda_m, \mathrm{u}_\mathrm{m})$ are the 'm' eigen-esteem cross product sets of the variance lattice 'B'. At that point, select the most noteworthy p eigen-esteems that pays supplementary to the prediction of the categorical data whereas the rest of the m–p esteems are of shallow importance with commotional information. The diminished subarea may be determined utilizing the Eq. (11)

$$\frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \geq S \tag{11}$$

Here 'S' is the proportion of variety in the diminished subarea to the overall variety in the upper proportional area. Accordingly, it gets MXP lattice Y_i comprising the p eigen cross products in the tuples. The information signified through the principal highlights obsessed by the decreased p proportional subarea as per Eq. (12).

$$Y_i = U^t \Phi_i = U^t (X_i - \mu) \tag{12}$$

3.1.5 Linear discriminant analysis (LDA)

LDA is a conventional measurable methodology for classification-based proportionality decrease and categorization. It figures an ideal change (prediction) by limiting the inside class separation and boosting the outside class

separation at the same time, in this way accomplishing most extreme class segregation. The ideal change in LDA may be promptly processed by means of implementing an eigen disintegration on the supposed disperse lattices. LDA has been utilized generally in numerous applications including from top to bottom proportional information [62].

LDA discovers the cross products in the basic space which discriminates greatest between the classes [63]. Aimed to the entire instances of every classes, the outside-class disperse lattice S_b and the inside class disperse lattice S_w are characterized as:

$$S_b = \sum_{i=1}^{c} (\mu_i - \mu)(\mu_i - \mu)^T \tag{13}$$

$$S_w = \sum_{i=1}^{c} \sum_{j=1}^{M_i} (Y_j - \mu_i)(Y_j - \mu_i)^T \tag{14}$$

Here M_i is the count of preparing instances in class i, c is the count of different classes, μ_i is the average cross product of instances having a place with class i and Y_j signifies to the collection of instances having a place with class i by means of Y_j being the jth information of the categorical output. S_w signifies the spread of the attributes about the average of each categorical output and S_b signifies to the spread of attributes about the complete the average of each categorical output. The objective is to amplify although limiting S_b, at the end augment the proportion

$$\det|S_B| / \det|S_w \tag{15}$$

This proportion is amplified once the section cross product of the prediction lattice is the eigen cross products of $S_w^{-1} S_b$. So as to avoid S_w to get solitary, IG is utilized as a pre-processing phase.

3.2 ML classifiers

Interruption identification is practiced by grouping the network traffic data into the legitimate one or the malicious one. There are numerous categorical outputs predicting methods are existing, however ML algorithms have included novel period for obscure samples categorization to advance the identification percentage [64]. Some of the ML classifiers are deliberated in this segment.

3.2.1 Naïve bayes (NB)

NB is the utmost well-known ML technique meanwhile from the past. Its straightforwardness makes the system appealing in various applications and sensible exhibitions are accomplished in the applications in spite of the fact that the method of learning depends on an idealistic individual presumption [65]. The NB classifier normally utilize Bayes' standard:

$$p(a_i|b) = \frac{p(a_i)p(b|a_i)}{p(b)} \tag{16}$$

Here, $p(a_i|b)$ is the back likelihood of class a_i specified another record b, $p(a_i)$ is the likelihood of class a_i that can be determined with:

$$p(a_i) = \frac{M_i}{M} \tag{17}$$

Here, M_i is the count of records allocated to class a_i and M is the count of classes, $p(b|a_i)$ is the likelihood of a record b specified a class a_i and $p(b)$ is the likelihood of report b.

3.2.2 Support vector machine (SVM)

SVM is an opportunity based rectilinear algorithm that develops the excitable levels in an upper proportional area as shown in Eq. (18).

$$f(x) = \beta_0 + \beta^t x \tag{18}$$

Here β is recognized as the mass cross product and β_0 as the predisposition. Through escalating β and β_0 an ideal level may be developed. Let $|\beta + \beta_0| = 1$ be an excitable level and the learning cross products nearest to the learning level are known as the SVM model cross products. The separation among a spot x and the excitable level is meant via

$$\frac{|\beta + \beta t x|}{||\beta||} \tag{19}$$

In this way, the separation $\frac{1}{||\beta||}$ must be limited to acquire ideal excitable level in the prediction of categorical outputs [66].

3.2.3 J48

One of the most popular kinds of decision tree algorithm is J48 algorithms. It is the modernized variant of C4.5 algorithm and depends on ID3 algorithm [67]. It utilizes the idea of data entropy aimed at constructing decision tree using the set of observations and informational index. It utilizes the way in which every attribute of the information may be utilized to settle on a result by dividing the information into minor subgroups that comprise of hubs that creates an established tree. The decision tree has three kinds of hubs. Primary hub is an origin hub that has no inbound limits, the subsequent hub is the internal hubs (divisions) and every single other hub are known as leaves otherwise called end or choice hubs, every leaf hub is given a class name, and other inner hubs comprise attribute

assessment constrains to isolate instances that have various attributes [67].

For constructing decision tree, the algorithmic steps are composed beneath.

1. Verify the existence of the parent hubs.
2. By dividing every feature (or) attribute, A, identify the information index for every A.
3. Let A' will be the superlative attribute with the maximum information index.
4. Generate a result hub that divides the attribute A'.
5. Recurrences on the sub records acquired via dividing a finest attribute in the hub and include those hubs as offspring hubs.

3.2.4 Random forest (RF)

Breiman has familiarized the RF algorithm [68]. The RF strategy is built utilizing the assortments of delicately-associated decision trees. A launch model of the set of observations is utilized to prepare every decision tree in the RF model. The finest partition is picked at every hub from an arbitrary subgroup of the attributes. This strategy ensures that every DECISION TREE utilizes individual attributes from the set of observations. Subsequently, it diminishes the measurable relationships on the remainder of the decision trees. RF is implemented in dual step, initial step is to make the RF by joining N DECISION TREE, and the final step is to create expectations for every decision tree made in the primary stage.

The steps for implementing RF algorithm is depicted beneath:

1. Pick arbitrary I information spots from the set of observations.
2. Construct the decision tree related with the chosen information spots (Subgroups).
3. Select the numeral J for decision trees, on which the model needs to be assembled.
4. Iterate the Step 1 and 2.
5. For original information spots, discover the forecasts of every decision tree, and allocate the original information spots to the classification which successes the dominant part of the divisions.

3.2.5 Multi-layer perceptron (MLP)

An MLP is a feed-forward artificial neural network (ANN) framework which assigns a collection of inlet information against a collection of suitable outlets. It comprises of several tiers of hubs in a digraph, by means of every tier completely associated with the following one. Excluding the inlet hubs, every hub is a preparing component through a curvilinear stimulation work [69]. MLP utilizes gradient decent based back propagation for preparing the system. This class of systems comprises of numerous tiers of calculation components, typically interrelated in a forward direction. In numerous appliances the components of these systems imply a sigmoid stimulation as an actuation work. It is the greatest ordinary ANN framework and aims to estimate a model f ().

Aimed at the samples, an activation function $y = f * (x)$ which assigns an inlet x to outlet label y, the MLP identify the greatest estimation of that activation function by characterizing a plotting, $y = f(x; \theta)$ and training the finest limits θ meant for it. The MLP systems are made out of numerous capacities that are bounded simultaneously. A system by means of three capacities or tiers would frame $f(x) = f(3)(f(2)(f(1)(x)))$. Every tier is made out of components that play out a relative change of a direct summation of the data sources. Every tier is denoted by means of $y = f(WxT + b)$, in which f is the stimulation work, W represents the list of limits, or loads, in the tier, x denotes an inlet cross product, it may also be the outlet of the past tier, and b indicates the one-sided cross product. The MLP tiers comprise of a numerous completely associated tiers, since every component in a tier is associated with the entire components in the past tier. In a completely associated tier, the limits of every component are free from the remaining components in the tier, which implies every component have an interesting group of loads.

3.2.6 k-Nearest neighbour (kNN)

The kNN algorithm is a sample dependent algorithm that anticipate with respect to the set of observations of training records that are same as the trial record. Therefore, it doesn't construct an obvious descriptive framework for the set of observations c_i [70]. A sample is categorized by means of an equivalence dependent choice of its adjacent samples, through the sample being grouped to the category, which is the extremely familiar between its k adjacent samples, where k is an optimistic value. In the event k = 1, at that point the sample is essentially allocated according to the group of its adjacent sample. Specified a trial record d, kNN identifies the k adjacent samples between the training records. The equivalence sum of every adjacent sample record to the trial record is utilized as the bias of the categories of the adjacent records. The biased aggregate in kNN categorization is calculated as follows:

$$Weighted\ sum(d_j, c_i) = \sum_{d_j \in KNN(d)} sim(d, d_j)\delta(d_j, c_i) \quad (20)$$

The term kNN (d) denotes the record collection of k adjacent samples of record d. Whenever d_j has a place with

c_i, $\delta(d_j, c_i)$ rises to 1, or in any case 0. A trial record d, ought to have a place with the category that has the most elevated bias score.

4 Proposed methodology

A brief overview of the AWID data collection with an emphasis on the AWID-ATTCK-REDUCED (AAR) subgroup and correspondingly an outline of the proposed methodology has been explained in this segment. Figure 2. Represents the detailed working flow of the proposed system, that includes the following parts: wireless network dataset collection, data pre-processing and attribute preparation, optimum attribute set selection, dataset splitting into training and testing set, detection and classification of normal wireless network data and attack network data along with its attack types and evaluation of the proposed model. The workflow of the detection model is as follows:

(1) Wireless network data collection—Reduced samples of wireless network packets are collected from AWID dataset.
(2) Data pre-processing and attribute preparation
    1. Initial step includes the cleaning of noisy data in the AWID dataset.
    2. Pre-processing module converts the non-numerical values to numeric values, resulting vector represents 154 extracted attributes.
    3. Normalize the attribute values between the range [0,1] using min–max normalization.
    4. In the attribute preparation process string attributes are removed.
(3) Optimum selection of attributes.
    1. Random initialization of input training sample of attributes.
    2. Normalized training samples are fed to five different attribute selection methods namely, IG, CFS, CH, PCA, and LDA. Each attribute reduction method is executed individually. The subset of attributes reduced by each attribute selection method is collected separately.
    3. Repeat the process until each attribute selection method is trained to meet the conditions of iteration or the deviation condition.
(4) Splitting of dataset—The pre-processed dataset is divided into training and testing set before performing the classification process. The data in the training set is used by the ML algorithms in the learning process of the IDS model, whereas the testing set is used in the evaluation process.

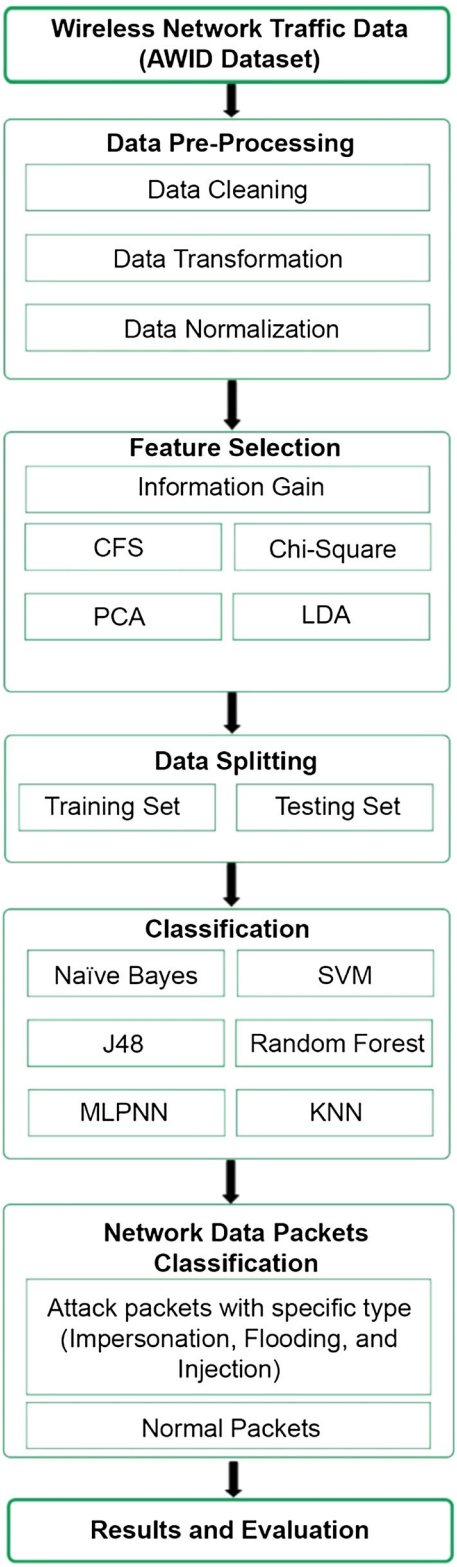

Fig. 2 Structure of WIDS against wireless attacks utilizing different attribute reduction and ML techniques

(5) Detection and classification of wireless network data.

1. The training set of data is used individually by six different ML algorithms namely, SVM, NB, RF, J48, kNN, and MLP.
2. Each ML algorithm is executed separately by including the reduced set of attributes obtained by each attribute reduction techniques separately.
3. Repeat the process for all the six ML algorithms to be executed with all the five attribute reduction techniques.
4. After training the IDS model, the wireless network data are classified into normal wireless network data and attack network data.

(6) Evaluation of the experiment—The wireless network packets are detected into normal and attack packets and then the overall accuracy, precision, recall, F-measure, True Positive Rate (TPR) and False Positive Rate (FPR) for each ML algorithms along with the reduced attribute sets are calculated and evaluated using the testing set.

4.1 Dataset collection and dataset description

The analyses of proposed work have been carried out utilizing an open data collection. AWID [56] is the marked data collection, that has its own place in the wireless sector and comprises a huge combination of legitimate and malicious wireless data instances. It incorporates four sections, together with two diminished data collections specifically, AWID-CLASS-REDUCED (ACR) and AAR for the study involved in Wireless WIDSs, and two complete data indexes to be specific, AWID-CLASS-FULL (ACF) and AWID-ATTACK-FULL (AAF) for big-data researches.

The two diminished informational indexes comprise of four categories namely, impersonation, injection, flooding, normal wireless data instances and fifteen categories namely, Evil twin, Hirte, Probe Request, Amok, Disassociation, Deauthentication, Beacon, Arp, Cafe latte, Fragmentation, Request to send (Rts), Clear to send (Cts), Power saving, normal wireless data instances. The count of learning instances of every diminished informational index is 1,795,575, and the count of trial instances is 575,643 accompanied by 92% of legitimate category instances. The count of attributes is 154, indicating the WLAN frame fields and the category name is the end field.

4.2 Pre-processing

In real world, data are generally dirty such as it contains errors, missing value, duplicate, outlier, incomplete, irrelevant and inconsistent data. The reason for information pre-processing is to remove the commotion information, extricate attributes, and changes the initial information into a structure which will be further efficiently and successfully handled with the end goal of the client. The pre-processing procedures are fundamental and significant in IDS because of the various forms of wireless network traffic instances, that have various kind of configurations and measurements. The following subsections, gives a comprehensive explanation of these techniques that are utilized in the proposed study. This strategy assists with improving the effectiveness of the ML classifiers in the classification of the information accurately.

The data preprocessing involves three stages: Data Cleaning, Data Transformation and Data Normalization. The working procedure of these methods are explained below.

4.2.1 Data cleaning

This stage is responsible for removing any records containing a missing values and inconsistent values. It also removes duplicate records in the data set. A portion of the attributes are essential for interruption identification, while some attributes can turn as a commotion; producing a damaging effect on the training rate & the precision. In this way, the analysis removes the unessential attributes afore progressing to the subsequent stage. Also, the data collection comprises of the symbols like "?" for inaccessible esteems for the consequent attributes. In this step, the symbols are allotted to nil esteem [71].

4.2.2 Data transformation

Subsequent to information cleaning the following phase of pre-processing is to change or alter the attributes that have content structures to numeric structure to be appropriate for ML algorithms. In AWID, the SSID attribute is of string form. Several attributes are of numeric form. String attributes are assigned to numeric esteems by including a lower estimation of 1 and a higher estimation of N, here N is the count of strings. A few hexadecimal attributes are changed into whole numbers. This change was used on the succeeding attributes: radiotap.present.reserved, wlan.fc.-type.subtype, wlan.fc.type, wlan.fc.dc, wlan.ra, wlan.da, wlan.ta, wlan.ra,wlan.sa, wlan.bssid, wlan.mgt.fixed.capabilities, wlan.mgt.fixed.listen.ival, wlan.wep.iv, wlan.-wep.key, wlan.qos.ack. Four significant class marks are

assigned to 4 numeric esteems from 1 to 4, as 1 = normal class instances, 2 = impersonation class instances, 3 = flooding class instances, and 4 = injection class instances.

4.2.3 Data normalization

The scope of estimations of unique information might be extraordinary. For ML classifiers, the method that is involved probably may not work usually in exclusive of attribute scaling. For example, Euclidean separation among two hubs ought to be determined in kNN classifier. In AWID, the scope of information of certain segments is tremendous. The difference of separation is chosen by information from those measurements. Because of the details referenced above, information must be standardized (or) normalized. Min–max normalization is utilized to create all the estimations of information to fall in the range [0,1], as represented in the Eq. (21).

$$X = \frac{X - X_{MIN}}{X_{MAX} - X_{MIN}} \tag{21}$$

Here X indicates whole number estimation of every class instance of the attribute index, X_{MAX} represents the highest estimation of the attribute index X, and X_{MIN} shows the minimal estimation of the attribute index X.

4.3 Attribute reduction

Attribute reduction is a procedure to choose the most appropriate attribute list by eliminating unrelated or repeated attributes. Only subsets of original attributes are selected. Attribute Reduction process comprise of succeeding stages:

1. *Attribute selection* This stage is utilized to create a list of attributes from the complete attribute set.
2. *Attribute estimation* This stage is utilized to evaluate the appropriateness of the reduced attributes depending on the resulted set of attributes.
3. *Attribute validation* This stage is utilized to evaluate whether the reduced attributes are legitimate or not.

The proposed methodology uses five attribute reduction techniques namely, IG, CFS, CH, PCA, and LDA. Wholly these attribute reduction techniques are utilized to calculate a value for every single attribute and therefore a predefined count of attributes are chosen in the categorization procedure according to the positioning got from the calculated value as appeared in Fig. 3. The working process of each attribute reduction technique is explained in detail in the Section III.

Figure 3 depicts the working procedure of the attribute reduction process. Initially, this process starts with the pre-processed set of AWID data. The pre-processed AWID data includes the original set of attributes. Each attribute reduction algorithms make use this original set of attributes for selecting an optimum set of attributes. According to the Fig. 3. IG is the first attribute selection technique implemented in the processed system. The IG algorithm calculates an information gain ratio for each attribute. Based upon the likelihood with the class labels, the attributes with higher information gain ration values are selected. This process is repeated until the arises a deviation between the likelihood of attributes and the class labels. Following the IG algorithm, CFS attribute reduction is implemented. In CFS technique, a correlation value is evaluated between -1 and 1 based on the recurrence of each attribute for each class label. The attributes with higher correlation values are reduced from the original attribute set. CFS algorithm is executed until the deviation of correlation values arises.

Next to the CFS technique, CH attribute selection algorithm is implemented. It makes use of the recurrence count values of each attribute. This process is repeated till the recurrence count value reaches nil. Consequently, PCA algorithm calculates the values of mean and variance of the attributes. Based on the mean and variance values, a co-divergence matrix is constructed. Then using the top-down approach PCA eliminates the least significant attributes. Finally, LDA algorithm is implemented using the cross product mean value. LDA algorithm calculates the cross product of each attribute and obtain its mean value. It eliminates the attributes with least cross product mean values. According to Fig. 3, each algorithm is executed individually in order to avoid the ambiguity in the attribute selection process and it separately records the optimum set of attributes obtained by each algorithm. The attribute set obtained by each attribute selection technique is listed in the Sect. 5 with detailed explanation. The reduced attribute set is used by the classification algorithms in the following section for the process of attack detection and classification.

4.4 Classification

In this examination, six ML techniques namely, SVM, NB, RF, J48, kNN, and MLP are utilized in Wireless attacks identification. These techniques are utilized as a result of the easiness, efficiency, and exactness. The operating method of every technique are described briefly in the Segment III. For the categorization procedure, the pre-processed information is partitioned into dual sections i.e., collection of training samples (or) instances (80%) and collection of testing samples or instances (20%).

In the Fig. 4, the wireless network samples are gathered and preprocessed in the training stage in order to remove the irrelevant attributes. The entire set of five attribute

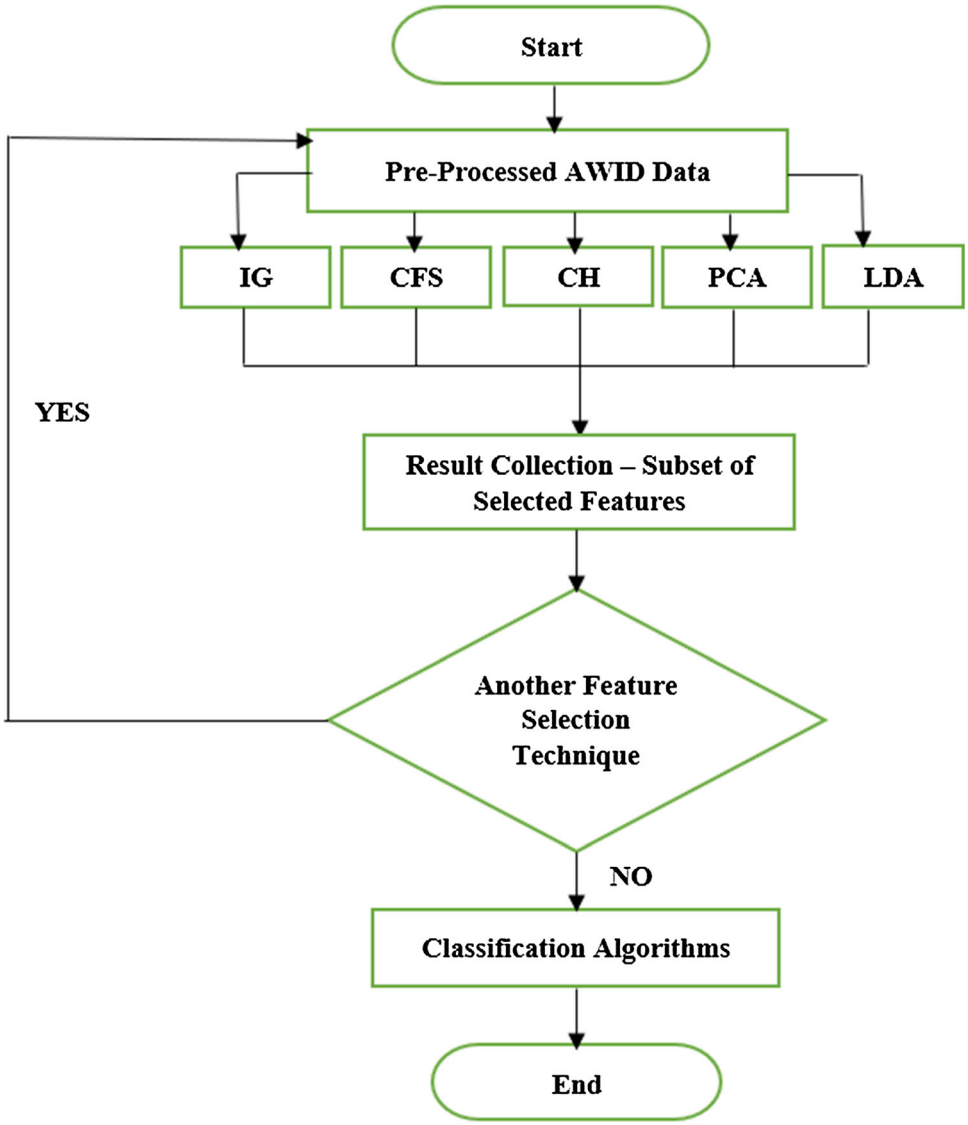

Fig. 3 Flow procedure of attribute reduction process

reduction techniques are implemented on the pre-processed information in order to reduce the high-level instances to the low-level instances. The pre-processed instances and reduced feature sets are utilized in training process to construct a fundamental model of legitimate wireless network instances along with malicious wireless network instances. In testing stage too, the trial instances are pre-processed to decrease the measurement from maximum to minimum categorization utilizing previously declared five attribute reduction methods. Processed test instances are related with the reference model constructed through the training stage and therefore the instances are categorized consequently.

As per the Fig. 4, the proposed WIDS used six ML algorithms in a sequence to implement a ML model. Every ML technique accepts a network trace as input and categorizes it as normal, flooding, or unified impersonation and injection traffic class. If the network trace is categorized as the unified class by one of the ML techniques, at that point it will be assessed by the resulting techniques; in any case, the forecast will be straightforwardly detailed as a yield of the WIDS. In such a way all the six classifiers are involved in the classification of the wireless network data. The trained ML model is evaluated using the testing test of data.

5 Experimental result analysis

This segment presents the major results obtained from the proposed experimentations. The experimental evaluation is carried out in the Weka, Rstudio and Anaconda Navigator Python tools, in which various combination of attribute reduction techniques and ML classifiers are implemented

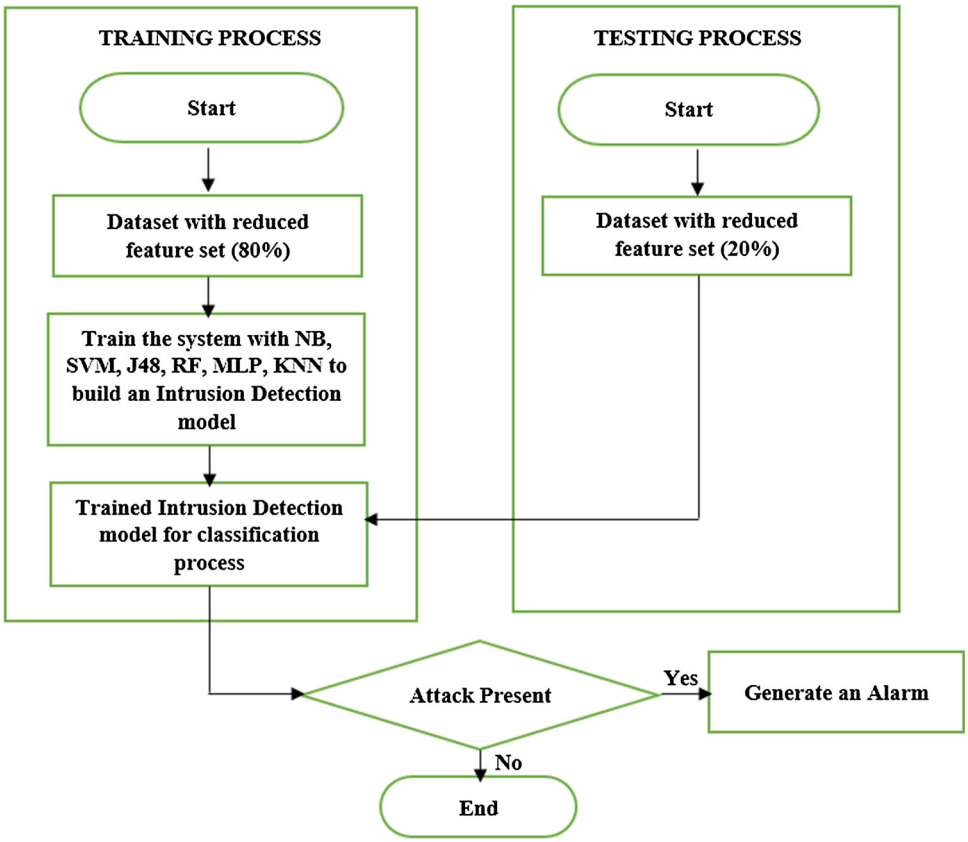

Fig. 4 Flow procedure of training and testing process under classification

on the input data, which was operating on a PC with Intel Core i7 3.30 GHz minimum of 16 GB RAM and 1 TB Hard Disk. The proposed model was prepared with the set of training instances and formerly estimated with the set of testing instances. The research is conducted on ACR training and ACR testing datasets.

The performance valuation of the research is incorporated by means of Accuracy, precision, detection rate (DR), and false alarm rate (FAR). The subsequent formulas are utilized to compute the above-mentioned performance metrics:

$$TPR\ or\ DR = \frac{TP}{TP + FN} = \frac{No.\ of\ correctly\ detected\ WiFi\ intrusions}{Total\ no.\ of\ WiFi\ intrusions} \tag{22}$$

$$\text{FPR } or\ FAR = \frac{FP}{TN + FP} = \frac{WiFi\ normal\ as\ WiFi\ intrusions}{WiFi\ intrusions} \tag{23}$$

$$True\ negative\ rate\ (TNR)\ or\ specificity = \frac{TN}{TN + FP} = \frac{Correct\ WiFi\ normal}{WiFi\ normal} \tag{24}$$

$$False\ negative\ rate\ (FNR) = \frac{FN}{TP + FN} = \frac{WiFi\ intrusions\ as\ WiFi\ normal}{WiFi\ intrusions} \tag{25}$$

$$\text{Accuracy} = \frac{TP + TN}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} = \frac{Correct\ classification\ of\ WiFi\ network\ instances}{All\ WiFi\ instances} \tag{26}$$

$$\text{Precision} = \frac{TP}{\text{TP} + \text{FP}} = \frac{Correct\ WiFi\ intrusions}{WiFi\ network\ instances\ classified\ as\ intrusions} \tag{27}$$

Here,

True positive (TP): Categorizing an interruption attack as an interruption attack.
False positive (FP): Inaccurately categorizing legitimate instance as an attack.
True negative (TN): Accurately categorizing legitimate instance as a legitimate instance.
False negative (FN): Inaccurately categorizing an attack as a legitimate instance.

In the proposed investigation, five traditional attribute reduction techniques are implemented individually on the original pre-processed set of 154 wireless network attributes. Table 1 shows the count of attributes reduced using each attribute reduction techniques and correspondingly list the counts of the reduced attributes. According to the findings in Table 1, IG & CH attribute reduction techniques reduce 154 attributes into 10 attributes, among which nine of them are common. CFS method selects five attributes, whereas PCA and LDA selects 25 and 22 attributes respectively. Both PCA and LDA produce 20 common attributes. In general, at the end of attribute reduction process, 20 attributes can be listed as common attributes. The 20 common attributes are listed in Table 2. These 20 attributes have been finalized, by analyzing the attributes reduced by each attribute reduction techniques individually.

In the proposed model before performing the categorization process with the attribute reduction techniques, the classification of AWID dataset without attribute reduction process have been performed using the classifiers NB, SVM, J48, RF, MLP, and kNN. The experiment has been conducted using 10-cross validation steeps for each ML algorithms and the results of the same has been showed in the Table 3 and Fig. 5 respectively. The classification process without attribute reduction methods has been carried out to analyze the variations in the performance metrics of the classifiers, along with the process of involving attribute reduction techniques.

From the Fig. 5, it is evident that RF classifiers perform with better accuracy of 90%, than other ML classifiers. NB classifier produces lower accuracy rate of 85.4%, than other classifiers in the model. TPR is higher for RF, SVM, and J48 with 92.6%, 91.3% and 91.1% respectively, whereas the TPR is lower for kNN algorithm with 85.9% and FPR is lower for RF with 8.5% and higher for NB and kNN algorithms with 19.1% and 18.2% respectively, than other classifiers in the model. The precision and F-measure are higher for RF with 92% and 93.6% respectively and lower for NB with 85% and 86.1% respectively, than other classification algorithms. The CPU build in time varies for each classification algorithms. Among the six classifiers MLP has higher CPU build in time and kNN has lower CPU build in time with 20.1 s and 0.07 s respectively. Therefore, from the results it is evident that RF algorithm performs better than all other ML algorithms without using attribute reduction methods. However, the performance of all the six ML algorithms can be improved with attribute reduction techniques.

The proposed study examined the execution of the six ML algorithms NB, SVM, J48, RF, MLP, and kNN using the attribute reduction methods IG, CFS, CH, PCA, and

Table 2 List of common attributes reduced using five various attribute reduction techniques

S. no	Common attributes
1	frame.time_epoch (4)
2	frame.len (8)
3	radiotap.datarate (47)
4	radiotap.channel.type.cck (50)
5	wlan.fc.type (66)
6	wlan.fc.subtype (67)
7	wlan.fc.ds (68)
8	wlan.fc.pwrmgt (71)
9	wlan.fc.protected (73)
10	wlan.duration (75)
11	wlan.ra (76)
12	wlan.da (77)
13	wlan.ta (78)
14	wlan.sa (79)
15	wlan.bssid (80)
16	wlan.seq (82)
17	wlan_mgt.fixed.reason_code (110)
18	wlan.wep.iv (141)
19	wlan.wep.key (142)
20	wlan.wep.icv (143)

Table 1 Represents the no. of attributes reduced using each attribute reduction technique

Attribute reduction technique	No of attributes reduced from 154 attributes	Reduced attributes
IG	10	4, 5, 6, 7, 8, 9, 38, 75, 82, 154
CFS	5	4, 8, 47, 68, 71
CH	10	4, 5, 6, 7, 8, 9, 38, 81, 82, 154
PCA	25	4, 8, 11, 47, 50, 66, 67, 68, 71, 73 75, 76, 77, 78, 79, 80, 82, 83, 108, 110, 112, 123, 141, 142, 143
LDA	22	4, 8, 47, 50, 66, 67, 68, 71, 73, 75, 76, 77, 78, 79, 80, 82, 104, 107, 110, 141, 142, 143

Table 3 Overall performance of the six different classifiers without attribute reduction

ML algorithms	CPU time (sec)	Accurately classified instances (%)	Inaccurately classified instances (%)	TPR (%)	FPR (%)	Precision (%)	F-measure (%)
NB	2.1	85.4	14.6	88	19.1	85	86.1
SVM	8.8	89.1	10.9	91.3	11.7	90.4	89.2
J48	10.5	89.7	10.3	91.1	10.8	91.4	90.8
RF	7.67	90.0	10.0	92.6	8.5	92.0	93.6
MLP	20.1	88.4	11.6	86.3	20.3	89	89.5
kNN	0.07	88.0	12.0	85.9	18.2	90.2	89.7

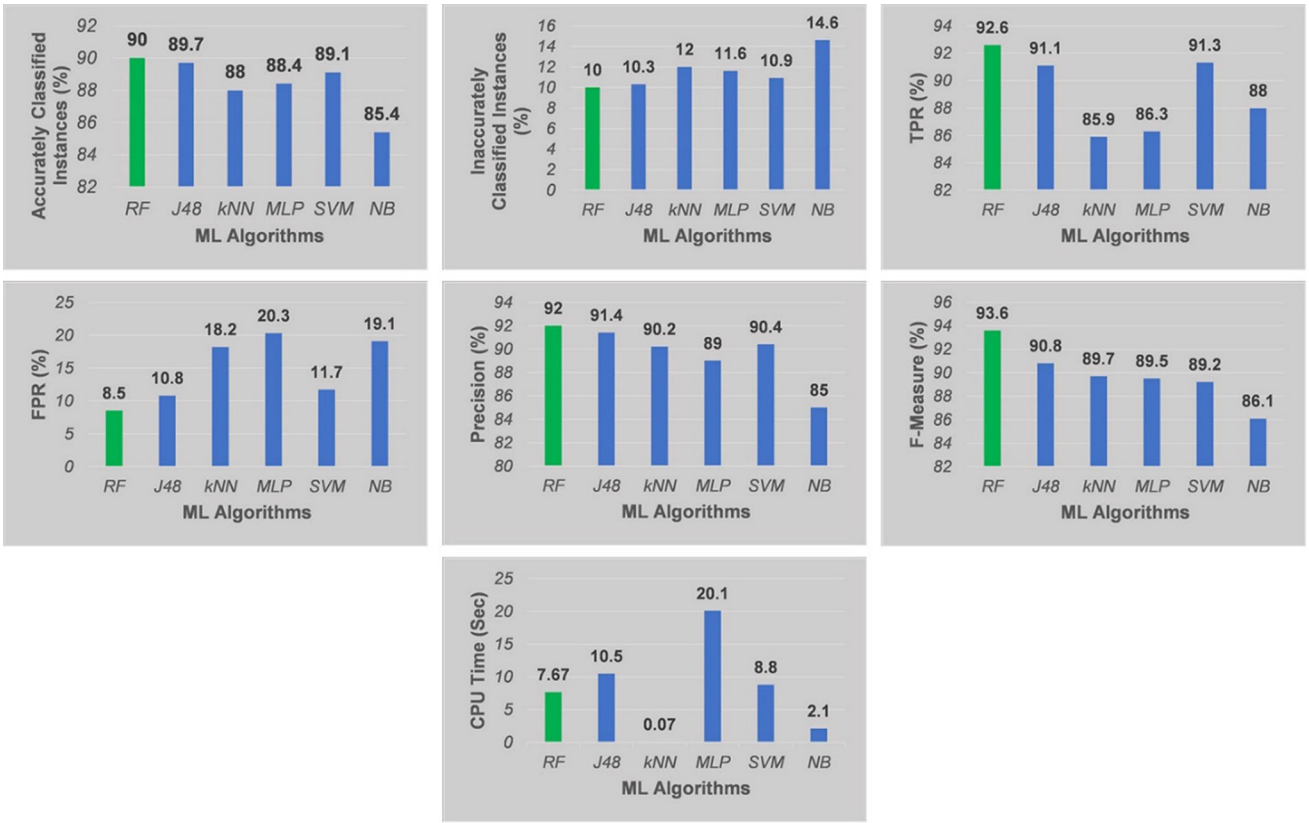

Fig. 5 Comparative results of different ML classifiers without attribute reduction

LDA. The summary of the results is depicted in Table 4. The complete description of the outcomes has been discussed below for each individual attribute reduction techniques, along with the ML algorithms.

Initially, the result of all six ML classifiers with the attributes selected by the IG attribute reduction method has been shown in Table 4. RF and J48 algorithms perform well with the accuracy of 94.8% and 94.4% respectively. Also, the Precision and F-measure are higher for RF and J48, than other classifiers. It is evident from the Table 4 that RF and J48 have lower FPR than other classifiers. kNN produces lower accuracy rate of 88.2%. NB, SVM, and MLP produces the accuracy of 91.2%, 93.1%, and 90.6% respectively. The CPU build in time is higher for MLP and lower for kNN with 15.9 and 0.07 s respectively. Thus, the results show that RF and J48 performs well with 10 attributes selected by IG attribute reduction method. The CPU build in time for RF and J48 is 4.22 and 6.47 respectively.

In the second stage, the result of all six ML classifiers with 5 attributes selected by the CFS method has been shown in Table 4. Only RF algorithm performs well with the accuracy of 94%. Also, the Precision and F-measure are higher for RF with 97.1% and 97% respectively, than other classifiers. It is apparent from the Table 4 that RF has lower

Table 4 Individual performance of six different classifiers with five different attribute reduction mthods

Attribute reduction techniques	ML algorithms	CPU time (sec)	Accurately classified instances (%)	Inaccurately classified instances (%)	TPR (%)	FPR (%)	Precision (%)	F-measure (%)
IG (10 reduced attributes)	NB	2.25	91.2	8.8	90.5	9.7	94	94.7
	SVM	3.44	93.1	6.9	93	7.4	93	94
	J48	6.47	94.4	5.6	93.1	7.0	98.3	98.1
	RF	4.22	94.8	5.2	93.8	6.2	99.0	99.4
	MLP	15.9	90.6	9.4	89.5	11.0	92.3	92
	kNN	0.07	88.2	11.8	85.8	14.9	87.6	87
CFS (5 reduced attributes)	NB	2.1	92.9	7.1	92	7.0	92	92.4
	SVM	2.9	91.6	8.4	90.8	9.5	90	90.3
	J48	5.45	93.4	6.6	93	7.3	95.2	95
	RF	3.96	94	6	93.9	6.9	97.1	97
	MLP	12.7	89.3	10.7	91.4	9.7	91	91.1
	kNN	0.06	84.5	15.5	88	11.6	89	89.5
CH (10 reduced attributes)	NB	2.25	91.2	8.8	90.5	9.7	94	94.7
	SVM	3.44	93.1	6.9	93	7.4	93	94
	J48	6.47	94.4	5.6	93.1	7.0	98.3	98.1
	RF	4.22	94.8	5.2	93.8	6.2	99.0	99.4
	MLP	15.9	90.6	9.4	89.5	11.0	92.3	92
	kNN	0.07	88.2	11.8	85.8	14.9	87.6	87
PCA (25 reduced attributes)	NB	3.67	89.1	9.9	87	13.2	88.6	88
	SVM	5.9	88.2	11.8	89.4	12.9	88.7	88.1
	J48	7.18	90.6	9.4	91.1	9.4	91.7	91.4
	RF	9.74	93.2	6.8	92.9	8.4	93.1	93
	MLP	19.4	88.9	11.1	89	10.7	89.5	89.1
	kNN	1.09	87.3	12.7	88.8	11.6	89.6	89.2
LDA (22 reduced attributes)	NB	3.3	88.7	11.3	87	12.9	88.1	88.4
	SVM	5.1	88.1	11.9	89.1	12.5	88.7	88.3
	J48	6.9	91.6	8.4	91	9.3	91.2	91
	RF	9.3	93.5	6.5	92.7	8.1	93.4	93.1
	MLP	18.8	88.9	11.1	89	10.7	89.5	89.1
	kNN	1.04	87.9	12.1	88.1	11.1	88.6	88.2

FPR of 6.9%, than other classifiers, also NB classifier has nearly lower FPR of 7% as RF classifier. Similarly, as with IG, kNN produces lower accuracy rate of 84.5% with CFS method. NB, and SVM produces the accuracy of 92.9%, and 91.6% respectively. The CPU build in time is higher for MLP and lower for kNN with 12.7 and 0.06 s respectively. Thus, the results show that RF performs well with CFS method. The CPU build in time for RF is 3.96.

In the third stage, the result of all six ML classifiers with 10 attributes reduced by the CH attribute reduction technique has been shown in Table 4. Since, both IG and CH attribute reduction methods have similar set of attributes, the results obtained are same for CH method as like IG method. RF and J48 algorithms perform well with the accuracy of 94.8% and 94.4% respectively. Also, the Precision and F-measure are higher for RF and J48, than other classifiers. It is evident from the Table 4 that RF and J48 have lower FPR than other classifiers. KNN produces lower accuracy rate of 88.2%. NB, SVM, and MLP produces the accuracy of 91.2%, 93.1%, and 90.6% respectively. The CPU build in time is higher for MLP and lower for KNN with 15.9 and 0.07 s respectively. Thus, the results show

that RF and J48 performs well CH attribute reduction method. The CPU build in time for RF and J48 is 4.22 and 6.47 respectively.

In the fourth stage, the result of all six ML classifiers with the attributes selected by the PCA attribute reduction method has been shown in Table 4. Along with PCA, RF algorithm performs well with the accuracy of 93%. Similarly, the Precision and F-measure are higher for RF with 93.1% and 93% respectively, than other classifiers. It is obvious from the Table 4 that RF has lower FPR of 8.4%, than other classifiers. Like with other attribute reduction methods, KNN produces lower accuracy rate of 87.3% and also MLP produces lower accuracy rate of 88.9%. NB, and SVM produces the accuracy of 89.1%, and 88.2%respectively, which are lower, when comparing with other attribute reduction methods. The CPU build in time is higher for MLP and lower for KNN with 19.4 and 1.09 s respectively. Thus, the result shows that RF performs well with 25 attributes selected by PCA attribute reduction method. The CPU build in time for RF is 9.74, which is higher than with other attribute reduction methods.

Finally, the result of all six ML classifiers with 22 attributes selected by the LDA attribute reduction method has been shown in Table 4. RF algorithm performs well with the accuracy of 93.5%. Also, the Precision and F-measure are higher for RF with 93.4% and 93.1% respectively, than other classifiers. It is evident from the Table 4 that RF has lower FPR of 8.1%, than other classifiers. MLP and KNN produces lower accuracy rate of 88.9% and 87.9% respectively. NB, and SVM produces the accuracy of 88.7%, and 88.1%respectively, which are lower, when comparing with other attribute reduction methods. The CPU build in time is higher for MLP and lower for KNN with 18.8 and 1.04 s respectively. Thus, the result shows that RF performs well with LDA attribute reduction method. The CPU build in time for RF is 8.1.

Thus, from the above discussion, it is evident that RF ML classifier performs well with all the five attribute reduction methods with higher accuracy rate and lower FPR, than other classifiers. Specifically, RF performs better with IG and CH attribute reduction techniques. Through all the attribute reduction methods, KNN classifier produces lower accuracy rate. MLP is the classifier, which takes longer time for execution with all the attribute reduction methods. Next to RF, classifiers like J48, NB and SVM performs well with decent accuracy rate. Among the five attribute reduction methods, PCA and LDA makes the classifiers to underperform than the other three attribute reduction methods. CFS attribute reduction method produces better results of all the classifiers with minimal number of attributes.

The overall performance of all the six ML classifiers with the attribute reduction process is shown in Table 5. Figure 6. shows that among the six classifiers, RF algorithm performs well with higher accuracy rate and lower FPR of 94.06% and 7.16% respectively. NB, SVM, and J48 algorithms also perform well, by producing a good percentage of 90.62, 90.82, and 92.88 respectively. The lower accuracy rate is produced by MLP and KNN with 89.66% and 87.22% respectively. Similar to RF, J48 also has lower FPR of 8%. Precision and F-measure is higher for RF and J48, than other algorithms. The CPU build in time is higher for MLP with 16.54 s and lower for kNN with 0.46 s. From the results it is evident that RF algorithm obtained better performance values under each measuring parameters than other ML algorithms with attribute reduction techniques. Thus, the accuracy of the RF algorithm is improved when combining it with the feature reduction techniques.

In Table 6, a few of the error measures are calculated for each ML classifiers. Lesser the error rate, more will be the accuracy and detection rate. Among the six classifiers, RF and J48 algorithms have lower mean absolute error rate of 0.025 and 0.067 respectively. Also, the root-mean square error is lower for RF classifier. Relative absolute error is lower for NB with value 0.34. The kappa statistics is higher for RF and J48 algorithms of values 0.99 and 0.99 respectively. MLP and KNN have higher mean absolute error, root-mean square error and relative absolute error. So, the accuracy for MLP and KNN algorithms will be lower and it is also evident from the Tables 4 and 5.

The combined results of six ML classifiers without and with attribute reduction methods has been depicted in the Table 7. From the table, it is evident that, the classifiers perform well with attribute reduction process in the detection of different attack classes, than without attribute reduction process.

The test accuracy of six ML classifiers without attribute reduction methods in the detection of different attack classes like, Normal, Impersonation, Flooding, and Injection are shown in Fig. 7. Among the six ML classifiers, RF detects normal, impersonation, flooding, and injection classes with higher accuracy of 91.9%, 91.4%, 90.1% and 89.9% respectively, than other classifiers. Next to RF, J48 and SVM algorithm performs well. Among the six ML algorithms MLP and kNN performs with lower accuracy rate in the detection of four class labels. The performance of the ML algorithms can be improved by combining it with feature reduction techniques. The following Fig. 8 depicts the increase in the performance of the ML algorithms combining with the feature reduction techniques.

The test accuracy of six ML classifiers with attribute reduction methods in the detection of different attack classes like, Normal, Impersonation, Flooding, and Injection are shown in Fig. 8. Among the six ML classifiers, RF detects normal, impersonation, flooding, and injection classes with higher accuracy of 95.9%, 95.1%, 94.7% and

Table 5 Overall performance of the six different ML algorithms with attribute reduction (attribute list with 10 to 32 attribute)

ML algorithms	CPU time (sec)	Accurately classified instances (%)	Inaccurately classified instances (%)	TPR (%)	FPR (%)	Precision (%)	F-measure (%)
NB	2.71	90.62	9.38	89.4	10.5	91.34	91.64
SVM	4.15	90.82	9.18	91.06	9.94	90.68	90.94
J48	6.49	92.88	7.12	92.26	8	94.94	94.72
RF	6.28	94.06	5.94	93.42	7.16	96.32	96.38
MLP	16.54	89.66	10.34	89.68	10.62	90.92	90.66
KNN	0.46	87.22	12.78	87.3	12.82	88.48	88.18

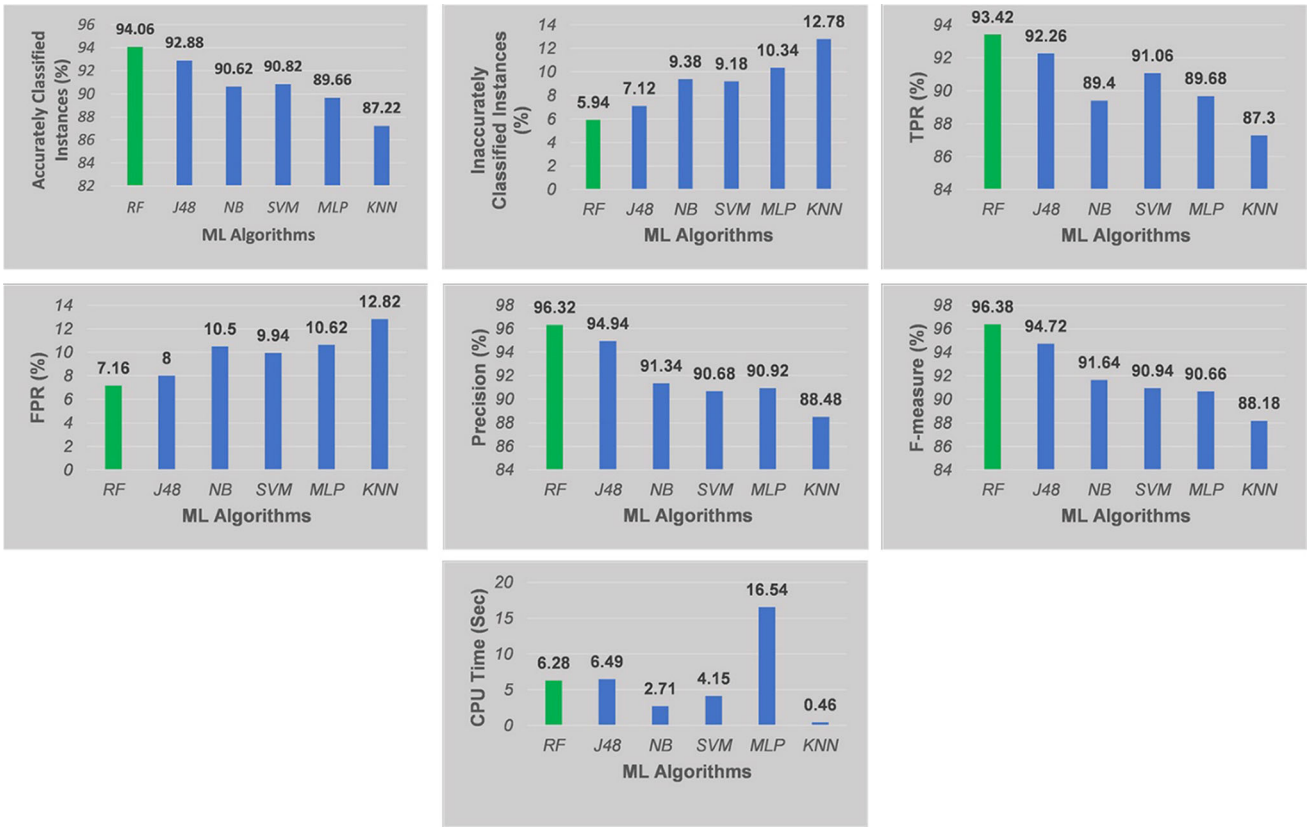

Fig. 6 Comparative Results of Different ML Classifiers with Attribute reduction

Table 6 Kinds of errors during testing

ML algorithms	Kappa statistics	Mean absolute error (MSE)	Root-mean square error (RMSE)	Relative absolute error
NB	0.86	0.965	0.305	0.34
SVM	0.90	0.546	0.471	0.55
J48	0.99	0.067	0.512	0.97
RF	0.99	0.025	0.038	0.506
MLP	0.70	0.887	0.672	0.87
KNN	0.45	0.991	0.853	0.736

Table 7 Test accuracy of six different classifiers for different classes of attacks with and without attribute reduction

ML algorithms	Class label	Test precision with 154 attributes (%)	Test precision with 10–32 attributes (%)
NB	Normal	86.3	91.1
	Impersonation	86	91.3
	Flooding	84.7	90.1
	Injection	85	89.6
SVM	Normal	90.3	91.7
	Impersonation	89.6	91.2
	Flooding	88.5	90
	Injection	88	89.4
J48	Normal	90.5	93.6
	Impersonation	89.8	93
	Flooding	87.2	92.1
	Injection	88.1	91.1
RF	Normal	91.9	95.9
	Impersonation	91.4	95.1
	Flooding	90.1	94.7
	Injection	89.9	93.8
MLP	Normal	89.9	90.8
	Impersonation	89.1	90.1
	Flooding	87.5	89.3
	Injection	86.9	89.1
KNN	Normal	89.8	88.5
	Impersonation	89.5	88.1
	Flooding	87.1	86.9
	Injection	86.7	86.4

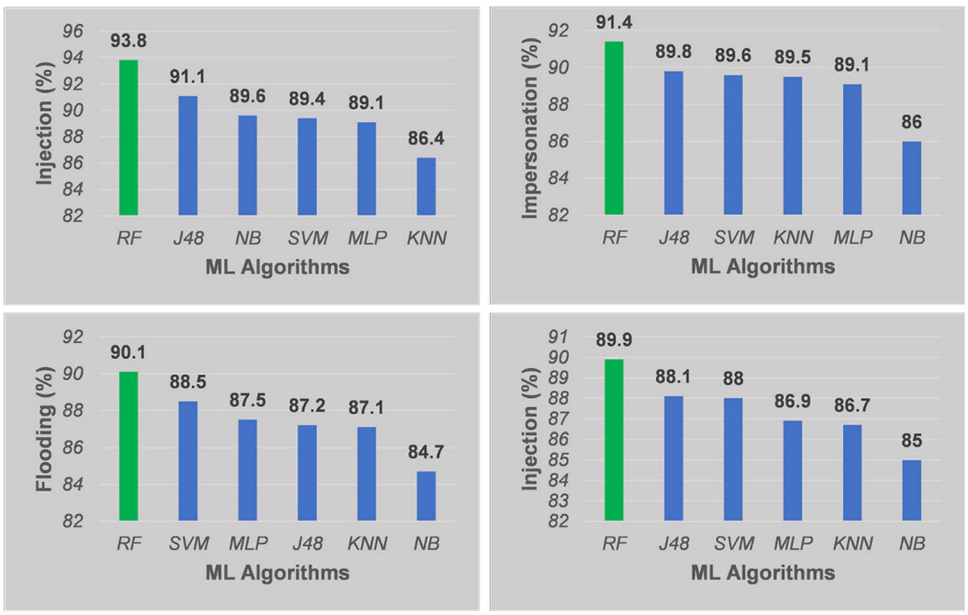

Fig. 7 Comparative results of six different classifiers for different classes of attacks without attribute reduction

93.8% respectively, than other classifiers. Next to RF, J48 algorithm performs well with higher accuracy rate of 93.6%, 93%, 92.1% and 91.1% for normal, impersonation, flooding, and injection classes respectively. Among the six ML algorithms MLP and kNN performs with lower accuracy rate in the detection of four class labels.

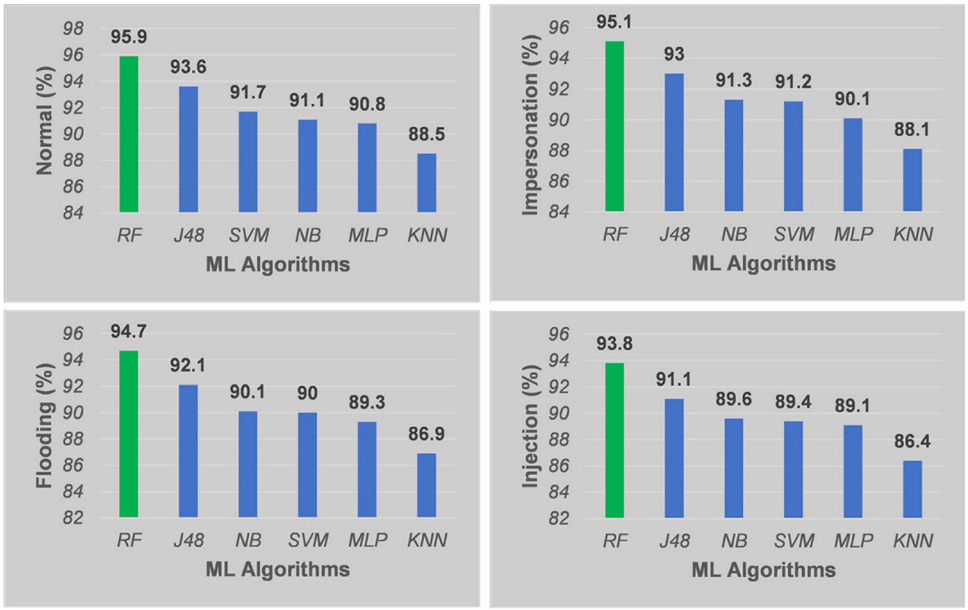

Fig. 8 Comparative results of six different classifiers for different classes of attacks with attribute reduction

6 Conclusion and future scope

The proposed IDS model depicts a broad investigation on the impact of five attribute reduction techniques on six ML algorithms in the identification of Wireless attacks. The fundamental commitment of this system is to adequately choose the suitable techniques for the involuntary attack identification in the Wireless network. The proposed IDS endeavor to figure out which attribute reduction technique and ML algorithm performs best for attack identification on Wireless network dataset. It additionally explores how attribute reduction techniques add in enhancing the categorization performance of the six ML algorithms on Wireless attack identification and categorization. The outcomes show that there is no predominant algorithm for every attribute reduction technique, besides there is no predominant attribute reduction technique for all the data collection capacities. The outcomes similarly show that utilizing the best five attribute reduction technique outputs enhanced outcomes associated and those attained utilizing the unique classifier, especially with the NB and SVM algorithms. Lastly, the outcomes show that the RF algorithm together with the IG and CH attribute reduction techniques accomplish the best execution during the Wireless attacks categorizing process, with a precision of 94%. Furthermore, the future scope will concentrate on building up an Intelligent IDS for identifying Wireless attacks in the wireless network data flow by investigating and implementing optimization algorithms like deep learning techniques, to resolve the dynamic attribute reduction issue for recognizing the Wireless attacks that happens in the real wireless network traffic flow.

References

1. Ray, S., Jin, Y., & Raychowdhury, A. (2016). The changing computing paradigm with internet of things: A tutorial introduction. *IEEE Design and Test, 33*(2), 76–96. https://doi.org/10.1109/MDAT.2016.2526612
2. Diechmann, J., Heineke, K., Reinbacher, T., & Wee, D. (2018). *The Internet of Things: How to capture the value of IoT*. Technical Report, 1–124. https://www.mckinsey.com/featuredinsights/internet-of-things/our-insights/the-internet-of-things-how-to-capture-the-value-of-iot#. Accessed 13 January 2021.
3. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks, 54*(15), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010
4. Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of Things: Vision, applications and research challenges. *Ad Hoc Networks, 10*(7), 1497–1516. https://doi.org/10.1016/j.adhoc.2012.02.016
5. Singh, S., & Singh, N. (2015). Internet of Things (IoT): Security challenges, business opportunities and reference architecture for E-commerce. In *International conference on green computing and Internet of Things (ICGCIoT)* (pp. 1577–1581). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/ICGCIoT.2015.7380718.
6. Weber, R. H. (2010). Internet of things: New security and privacy challenges. *Computer Law and Security Review, 26*(1), 23–30. https://doi.org/10.1016/j.clsr.2009.11.008
7. Kai, Z., & Lina, G. (2013). A survey on the Internet of Things security. In *Ninth international conference on computational intelligence and security* (663–667). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/CIS.2013.145.
8. Ioannis, A., Chrysostomos, C., & George, H. (2015). Internet of Things: Security vulnerabilities and challenges. In *IEEE symposium on computers and communication (ISCC)* (pp. 180–187). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/ISCC.2015.7405513.
9. Riccardo, B., Nicola, B., Vishwas, L., Alexis, O., & Alexandru, S. (2012). Secure communication for smart IoT objects: Protocol stacks, use cases and practical examples. In *IEEE international symposium on a world of wireless, mobile and multimedia*

networks (WoWMoM) (pp. 1–7). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/WoWMoM.2012.6263790.
10. Meidan, Y., Bohadana, M., Shabtai, A., Ochoa, M., Tippenhauer, N. O., Guarnizo, J. D., & Elovici, Y. (2017). *Detection of unauthorized IoT devices using machine learning techniques*. CoRR https://arxiv.org/abs/1709.04647.
11. Moskvitch, K. (2017). Securing IoT: In your smart home and your connected enterprise. *Engineering Technology, 12*(3), 40–42. https://doi.org/10.1049/et.2017.0303
12. Sivanathan, A., Sherratt, D., Gharakheili, H., Sivaraman, V., & Vishwanath, A. (2016). Low-cost flow-based security solutions for smart-home IoT devices. In *IEEE international conference on advanced networks and telecommunications systems (ANTS)* (pp. 1–6). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/ANTS.2016.7947781.
13. Kolias, C., Stavrou, A., Voas, J., Bojanova, I., & Kuhn, R. (2016). Learning Internet-of-Things security "hands-on." *IEEE Security and Privacy, 14*(1), 37–46. https://doi.org/10.1109/MSP.2016.4
14. Moustafa, N., Choo, K. K. R., Radwan, I., & Camtepe, S. (2019). Outlier Dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog. *IEEE Transactions on Information Forensics and Security, 14*(8), 1975–1987. https://doi.org/10.1109/TIFS.2018.2890808
15. Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *Computer, 50*(7), 80–84. https://doi.org/10.1109/MC.2017.201
16. Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., & Sheth, A. (2018). Machine learning for Internet of Things data analysis: Survey. *Journal of Digital Communications and Networks, 1,* 1–56. https://doi.org/10.1016/j.dcan.2017.10.002
17. AWID. (2014). http://icsdweb.aegean.gr/awid/features.html Accessed 25 February 2018.
18. Benzarti, S., Triki, B., & Korbaa, O. (2017). A survey on attacks in Internet of Things based networks. In *2017 International conference on engineering and MIS (ICEMIS)* (pp. 1–7). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/ICEMIS.2017.8273006.
19. Hussain, R., & Oh, H. (2014). On secure and privacy-aware sybil attack detection in vehicular communications. *Wireless Personal Communications, 77,* 2649–2673. https://doi.org/10.1007/s11277-014-1659-5
20. Dong, W., & Liu, X. (2015). Robust and secure time-synchronization against sybil attacks for sensor networks. *IEEE Transactions on Industrial Informatics, 11,* 1482–1491. https://doi.org/10.1109/TII.2015.2495147
21. Aslam, M., Ye, D., Hanif, M., & Asad, M. (2020). Machine learning based SDN-enabled distributed denial-of-services attacks detection and mitigation system for Internet of Things. In X. Chen, H. Yan, Q. Yan, & X. Zhang (Eds.), *Machine learning for cyber security. ML4CS 2020. Lecture notes in computer science 12486*. Cham: Springer. https://doi.org/10.1007/978-3-030-62223-7_16
22. Buddhika, T., & Pallickara, S. (2016). Neptune: Real time stream processing for internet of things and sensing environments. In IEEE *International parallel and distributed processing symposium (IPDPS)* (pp. 1143–1152). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/IPDPS.2016.43.
23. Hari, P. B., & Singh, S. N. (2019). Security attacks at MAC and network layer in wireless sensor networks. *Journal of Advanced Research in Dynamical and Control Systems, 11,* 82–89. https://doi.org/10.5373/JARDCS/V11I12/20193215
24. NSL-KDD. (2009). http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html. Accessed 31 January 2018.
25. KDD Cup 1999 Data. (1999). https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed 30 January 2018.
26. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *IEEE international conference on computational intelligence for security and defense applications (CISDA'09)* (pp. 53–58). IEEE Press, USA. https://doi.org/https://doi.org/10.1109/CISDA.2009.5356528.
27. Sabhnani, M., & Serpen, G. (2004). Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set. *Intelligent Data Analysis, 8*(4), 403–415. https://doi.org/10.3233/IDA-2004-8406
28. Liu, Y., Tian, D.-X., & Wei, D. (2006). A wireless intrusion detection method based on neural network. In *Proceedings of the second IASTED international conference advances in computer science and technology* (pp. 207–211), ACTA Press, USA.
29. Khoshgoftaar, T., Nath, S. V., Zhong, S., & Seliya, N. (2005). Intrusion detection in wireless networks using clustering techniques with expert analysis. In *Fourth international conference on machine learning and applications (ICMLA'05)* (pp. 6). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/ICMLA.2005.43.
30. Zhong, S., Khoshgoftaar, T. M., & Nath, S. V. (2005). A clustering approach to wireless network intrusion detection. In *IEEE international conference tools with artificial intelligence (ICTAI)* (pp. 196). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/ICTAI.2005.5
31. Boukerche, A., Machado, R. B., Juca, K. R. L., Sobral, J. B. M., & Notare, M. S. M. A. (2007). An agent based and biological inspired real-time intrusion detection and security model for computer network operations. *Computer Communications, 30*(13), 2649–2660. https://doi.org/10.1016/j.comcom.2007.03.008
32. Boukerche, A., Juc, K. R. L., Sobral, J. B., & Notare, M. S. M. A. (2004). An artificial immune based intrusion detection model for computer and telecommunication systems. *Parallel Computing, 30*(5), 629–646. https://doi.org/10.1016/j.parco.2003.12.008
33. Boukerche, A., & Notare, M. S. M. A. (2002). Behavior-based intrusion detection in mobile phone systems. *Journal of Parallel and Distributed Computing, 62*(9), 1476–1490. https://doi.org/10.1006/jpdc.2002.1857
34. Amiri, F., Yousefi, M. M. R., Lucas, C., Shakery, A., & Yazdani, N. (2011). Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications, 34*(4), 1184–1199. https://doi.org/10.1016/j.jnca.2011.01.002
35. El-Khatib, K. (2010). Impact of feature reduction on the efficiency of wireless intrusion detection systems. *IEEE Transactions on Parallel and Distributed Systems, 21*(8), 1143–1149. https://doi.org/10.1109/TPDS.2009.142
36. Schaffernicht, E., & Gross, H. M. (2011). weighted mutual information for feature selection. In T. Honkela, W. Duch, M. Girolami, & S. Kaski (Eds.), *Artificial neural networks and machine learning—ICANN 2011. ICANN 2011. Lecture notes in computer science, 6792*. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-21738-8_24
37. Kasliwal, B., Bhatia, S., Saini, S., Thaseen, I. S., & Kumar, C. (2014). A hybrid anomaly detection model using G-LDA. In U. Batra & A. Sujata (Eds.), *IEEE International advance computing conference (IACC)* (pp. 288–293). USA: IEEE Computer Society. https://doi.org/10.1109/IAdCC.2014.6779336
38. Sindhu, S. S. S., Geetha, S., & Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach. *Expert Systems with Applications, 39*(1), 129–141. https://doi.org/10.1016/j.eswa.2011.06.013
39. Stein, G., Chen, B., Wu, A. S., & Hua, K. A. (2005). Decision tree classifier for network intrusion detection with GA-based feature selection. In *Proceedings of the 43rd annual southeast*

regional conference (ACM-SE 43). (vol. 2, pp. 136–141). New York, NY, USA: Association for Computing Machinery. https://doi.org/https://doi.org/10.1145/1167253.1167288.
40. Sung, A. H., & Mukkamala, S. (2004). The feature selection and intrusion detection problems. In M. J. Maher (Ed.), *Advances in computer science—ASIAN 2004. Higher-level decision making. ASIAN 2004. Lecture notes in computer science*. Berlin, Heidelberg: Springer.
41. Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks, 174,* 1–21. https://doi.org/10.1016/j.comnet.2020.107247
42. Farahani, G. (2020). Feature selection based on cross-correlation for the intrusion detection system. *Security and Communication Networks*. https://doi.org/10.1155/2020/8875404
43. Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2006). Machine learning: A review of classification techniques. *Artificial Intelligence Review, 26*(3), 159–190. https://doi.org/10.1007/s10462-007-9052-3
44. Entezari-Maleki, R., Rezaei, A., & Minaei-Bidgoli, B. (2009). Comparison of classification methods based on the type of attributes and sample size. *Journal of Convergence Information Technology, 4*(3), 94–102.
45. Bakar, A. A., Othman, Z. A., Hamdan, A. R., Yusof, R., & Ismail, R. (2008). An agent-based rough classifier for data mining. In *Eighth international conference on intelligent systems design and applications (ISDA '08)* (vol.1, pp. 145–151). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/ISDA.2008.29.
46. Chebrolu, S., Abraham, A., & Thomas, J. P. (2005). Feature deduction and ensemble design of intrusion detection systems. *Computers and Security, 24*(4), 295–307. https://doi.org/10.1016/j.cose.2004.09.008
47. Li, Z., Li, Y., & Xu, L. (2011). Anomaly intrusion detection method based on k-means clustering algorithm with particle swarm optimization. In *International conference of information technology, computer engineering and management sciences* (pp. 157–161). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/ICM.2011.184.
48. Teng, S., Du, H., Wu, N., Zhang, W., & Su, J. (2010). Acooperative network intrusion detection based on fuzzy SVMs. *Journal of Networks, 5*(4), 475–483. https://doi.org/10.4304/jnw.5.4.475-483
49. Chen, W. H., Hsu, S. H., & Shen, H. P. (2005). Application of SVM and ANN for intrusion detection. *Computers and Operations Research, 32*(10), 2617–2634. https://doi.org/10.1016/j.cor.2004.03.019
50. Li, K. L., Huang, H. K., Tian, S. F., & Xu, W. (2003). Improving one-class SVM for anomaly detection. In *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No.03EX693)* (vol. 5, pp. 3077–3081). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/ICMLC.2003.1260106.
51. Ambwani, T. (2003). Multi class support vector machine implementation to intrusion detection. In *Proceedings of the international joint conference on neural networks* (vol. 3, pp. 2300–2305). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/IJCNN.2003.1223770.
52. Wang, J., Hong, X., Ren, R., & Li, T. (2009). A real-time intrusion detection system based on PSO-SVM. In *Proceedings of the international workshop on information security and application* (pp. 319–321).
53. Saxena, H., & Richariya, V. (2014). Intrusion detection in KDD99 dataset using SVM-PSO and feature reduction with information gain. *International Journal of Computer Applications, 98*(6), 25–29. https://doi.org/10.5120/17188-7369
54. Manekar, V., & Waghmare, K. (2014). Intrusion detection system using support vector machine (SVM) and particle swarm optimization (PSO). *International Journal of Advanced Computer Research, 4*(3), 808–812.
55. Huang, C.-L., & Dun, J.-F. (2008). A distributed PSO–SVM hybrid system with feature selection and parameter optimization. *Applied Soft Computing, 8*(4), 1381–1391. https://doi.org/10.1016/j.asoc.2007.10.007
56. Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2016). Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. *IEEE Communications Surveys and Tutorials, 18*(1), 184–208. https://doi.org/10.1109/COMST.2015.2402161
57. Abdulhammed, R., Faezipour, M., Abuzneid, A. A., & Alessa, A. (2018). Effective features selection and machine learning classifiers for improved wireless intrusion detection. In *International symposium on networks, computers and communications (ISNCC) C* (pp. 1–6). https://doi.org/https://doi.org/10.1109/ISNCC.2018.8530969.
58. Nguyen, H. A., & Choi, D. (2008). Application of data mining to network intrusion detection: Classifier selection model. In Y. Ma, D. Choi, & S. Ata (Eds.), *Challenges for next generation network operations and service management. APNOMS 2008. Lecture notes in computer science, 5297*. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-88623-5_41
59. Mukherjee, S., & Sharma, N. (2012). Intrusion Detection using Naive Bayes Classifier with Feature Reduction. In *Proceedings of the second international conference on computer, communication, control and information technology (C3IT)* (vol. 4, pp. 119–128). Elsevier—Procedia Technology. https://doi.org/https://doi.org/10.1016/j.protcy.2012.05.017.
60. Hall, M. A. (1999). *Correlation-based feature selection for machine learning*. PhD Thesis, University of Waikato, Hamilton, The New Zealand.
61. Jolliffe, I. T. (2005). *Principal component analysis*. Encyclopaedia of statistics in behavioural science. Hoboken: Wiley. https://doi.org/10.1002/9781118445112.stat06472
62. Ye, J. (2007). *CSE 494 CSE/CBS 598 (Fall 2007): Numerical linear algebra for data exploration—Two dimensional SVD and PCA*.
63. Delac, K., Grgic, M., & Grgic, S. (2005). Independent comparative study of PCA, ICA, and LDA on the FERET data set. *International Journal of Imaging Systems and Technology, 15,* 252–260. https://doi.org/10.1002/ima.20059
64. Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: practical machine learning tools and techniques* (3rd ed.). San Francisco, CA: Morgan Kaufmann Publishers Inc.
65. Khalifa, K., & Omar, N. (2014). A hybrid method using lexicon-based approach and naive Bayes classifier for Arabic opinion question answering. *Journal of Computer Science, 10*(10), 1961–1968. https://doi.org/10.3844/jcssp.2014.1961.1968
66. Shang-fu, G., & Chun-lan, Z. (2012). Intrusion detection system based on classification. In *IEEE international conference on intelligent control, automatic detection and high-end equipment* (pp. 78–83). IEEE Computer Society, USA. https://doi.org/https://doi.org/10.1109/ICADE.2012.6330103.
67. Upendra. (2013). An efficient feature reduction comparison of machine learning algorithms for intrusion detection system. *International Journal of Emerging Trends and Technology in Computer Science, 2*(1), 66–70.
68. Breiman, L. (2001). Random forests. *Machine Learning, 45,* 5–32. https://doi.org/10.1023/A:1010933404324
69. Spencer, M., Eickholt, J., & Cheng, J. (2015). A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM Transactions on Computational Biology and*

Bioinformatics, 12(1), 103–112. https://doi.org/10.1109/TCBB.2014.2343960 PMID:25750595.
70. Tan, S., & Zhang, J. (2008). An empirical study of sentiment analysis for Chinese documents. *Expert Systems with Applications, 34*(4), 2622–2629. https://doi.org/10.1016/j.eswa.2007.05.028
71. Larose, D. (2014). *Data preprocessing-discovering knowledge in data: An introduction to data mining* (pp. 27–40). Hoboken: Wiley. https://doi.org/10.1002/0471687545

M. Nivaashini received her B.E. degree in Computer Science and Engineering in 2015 from Anna University and the M.E. degree in Bio-metrics and Cyber Security in 2017 from Anna University. She is a Research Scholar in the Department of Computer Science and Engineering, at KPR Institute of Engineering & Technology, Coimbatore. She currently pursues her doctoral research (Ph.D.) in Intrusion Detection System against Wi-Fi attacks in Internet of Things. She published nearly 15 papers in International and National Journals and several papers in International and National Conferences. Her areas of interest include Artificial Intelligence, Internet of Things, Big Data Analytics and Cyber Security.

P. Thangaraj received his M.Sc. from Madras university (1983). He has obtained his M.E. (CSE) degree from Vinayaka Missions university (2007). He was awarded Ph.D. In Soft computing from Bharathiyar University (2004). He has 30 years of teaching experience at the college level. He worked as Dean at Department of computer applications, Kongu College of Engineering & Technology, Erode. He also worked as Dean at Department of Computer Science & Engineering, Bannari Amman Institute of Technology, Erode. Currently he is working as Professor and head of Computer Science & Engineering department at KPR Institute of Engineering & Technology, Coimbatore. He is guiding nine Ph.D. theses in these areas. He has published 70 papers in International and National Journals and also published around 25 papers in International and National Conferences conducted both in India and abroad. His area of interest is Wireless Networks, Soft Computing.