Open Access Article

AN OPTIMAL DEEP EARNING MODEL FOR TRAFFIC ANALYSIS AND FRAME-BY-FRAME ATTACK DETECTION IN WI-FI NETWORKS

V.S.Bharathidasan[1] , A. Prema Kirubakaran[2]

[1]Research Scholar, Vels Institute of Science, Technology & Advanced Studies(VISTAS), Chennai, India

[2]Associate Professor, Vels Institute of Science, Technology & Advanced Studies(VISTAS), Chennai, India

[1]dasan11181@gmail.com, [2]unjanai@gmail.com

Abstract: The term "Wi-Fi" refers to the wireless local area network that affords internet access in numerous familiar places, including stores, restaurants, cafes, and college campuses. The Internet's unavailability, which allows for many attacks, has delayed and sometimes completely disrupted these services, although they were made possible by this technology. Therefore, due to their inherent weakness, wireless interfaces have spurred the study and suggestion of traffic investigation and anomaly detection systems. This study introduces a scalable and modular algorithm structure for setting up deep learning (DL) based on a Convolutional Deep Belief Network (CDBN) with a genetic algorithm (GA) named OCDBN to detect malicious frames reliably and To detect attacks, with features optimized via a GA. This model uses a dissimilarity measure that deals with non-numerical and numerical features. It also uses the Aegean Wi-Fi Intrusion Detection (AWID) dataset to test how well the suggested Algorithm works. This work found up to 12 of the 14 attack classes in the AWID dataset recognized with high confidence just by looking at a single frame, as long as the appropriate features are perceived.

Keywords: Network attacks, Traffic classification, Wi-Fi Networks, Malicious traffic detection, Clustering, deep learning, and Genetic Algorithm

Introduction

Many Web users gain advantages from the growing power and reach of the Internet. However, as the number of people who rely on networks grows, so does the significance of keeping them safe. The network security goal is to stop potential access and change to resources like computers, networks, programs, data, etc. [1]. Nonetheless, as more and more financial, E-commerce, and defense operations are associated with the Internet, they are increasingly vulnerable to network attacks, which can cause significant losses. Providing efficient monitoring methods for and counteract attacks is crucial to keeping networks safe. And various attacks call for different responses. Thus, figuring out how to recognize new and previously unseen forms of network intrusion has become the most pressing problem in computer security.

Access to large and trustworthy datasets of Wi-Fi traffic streams is crucial for defining and testing effective offensive recognition and classification methods [2]. More significant investigation in this way is introduced. It concentrates on assessing Medium Access Control (MAC) gradient frames, besides using a wide-ranging evaluation of extracted features on all various areas and dimensions on

Received: January 03, 2023 / Revised: January 11, 2023 / Accepted: February 10, 2023 / Published: February 20, 2023

About the authors：V.S.Bharathidasan

Corresponding author- Email: dasan11181@gmail.com

collected frames; for example, [3] presents and analyses data obtained in an enterprise network. In [4], the authors make their most significant contribution by building a comprehensive test bed that permits collecting a large dataset called AWID that was meticulously produced and includes fourteen separate attacks commonly encountered in the Wi-Fi ecosystem.

This latter point is particularly intriguing because most attacks require the repeated delivery of frames by both the attacker and the defendant. Accordingly, the sequence of successive images is likely to matter. Research shows that even a single frame has enough data to identify most threats. Frames can reliably label as "good" or "bad," with "bad" indicating that such a frame is believed to remain a component of an attack [5]. The GA (genetic Algorithm) of classifications is crucial to this outcome because it tunes an appropriate parameterized distance measure capable of identifying a collection of essential features between many broad sets, which considers recorded network patterns in this connection. This research aims to investigate a novel technique for detecting assaults on Wi-Fi networks, one that uses a dynamically optimized classification scheme. In this context, it is helpful to extract features repeated features in Wi-Fi frames, which authenticate such attacks, by leveraging services offered by the AWID dataset [6] via a quality perceptions distance function. The resultant classifier is easily implemented, scalable, and comprises a modular design.

Studies have attempted a variety of machine learning (ML) techniques over the past few years to categorize network assaults without having comprehensive prior information on their properties. Unfortunately, due to their computational cost constraints, typical ML approaches cannot produce unique feature descriptors to characterize the attack detection problem. DL approaches, so-called for their architecture of deep layers to handle complex issues, represent a significant advancement in ML's ability to mimic the human brain with the structure of neural networks. This study is divided to explain the complexities of DL to people who want to study information security with DL (DL) methods but are intimidated by the subject matter. As a general rule, there is already quite a bit of examination on attack detection utilizing DL methods. The following is the work's most significant original contribution:

- Its primary goal is to train a classifier algorithm to determine whether or not a given MAC frame is associated with a particular attack type among a group of candidate assaults that have already been specified.
- Then, using an appropriate dissimilarity metric, demonstrating that it is possible to reliably detect all attack types in the AWID database with an accuracy of 95%;
- Finally, an OCDBN-based automated system with a modular and extensible architecture is proposed to determine the most pertinent traffic aspects to notice each attack.

Here is how the rest of the paper is laid out. Subsequently, a brief literature review in Section 2, this paper will introduce the AWID dataset and its primary properties and describe the assaults in Section 3. Section 4 describes the classifiers built with DL techniques and the employed processing strategy. Section 6 contains the experimental findings. Section 7 offers the final thoughts and discusses what comes next.

Related work

Numerous flow monitoring systems, outlier detectors, and malware categorization instruments have been built with heavy reliance on ML and DL techniques. To accomplish the goal of legitimate object recognition, as described by Zhu et al. [7], cameras using the Wi-Fi transmission mechanism are used to detect and establish a network connection. In particular, a detecting model is constructed by updating the cutting-edge SSD method to boost operational efficiency (Single Shot Detector). Nevertheless, the proposed method for moving objects is a challenging task due to the need for natural image synthesis.

Utilizing an adaptation process in a swarm of individuals, Dwivedi et al. [8] presented a multi-parallel adaptive evolutionary method that is tested on three IDS datasets such as AWID-ATK-R, NSL-KDD, and NGIDS-DS. Simulated annealing (SA) is then merged into a multi-parallel adaptive grasshopper optimization method to increase the quality of different agents after every iteration; this study has been a game-changer in the modern era of fast, accurate threat identification. Anomalous network threats can only be dealt with by combining traffic monitoring with strict security controls.

For their study, Agarwal et al. [9] zeroed in on flooding DoS attacks in wireless networks, in which a significant number of fake requests that appear to come from a good source are sent to an access point (AP) that has been compromised. When the AP must process a high number of counterfeit frames, it experiences a significant increase in load that can be described as a flooding Denial of Service assault. Throughout to detect flooding DoS (Denial of Service) threats in 802.11 networks, which are primarily undiscovered and can be costly to manage and maintain, this research develops a novel ML approach to detect in conjunction with an intrusion prevention system (IPS).

WIDPS (Wi-Fi Intrusion Detection and Prevention Systems) were developed by Mahini and Mousavirad [10] with a practical but straightforward correction factor for dealing with processes and finding transferred across an unsecured connection. Indeed, this model proposes an achievement assessment method for WIDPS based on the network application domain and the significance of secrecy, truthfulness, and accessibility as the multiple information security moralities security, as well as models the communication between the enterprises associated with the issue using game mechanics. However, there is a lack of study on how to develop best a complete model for evaluating the effectiveness of security systems in various settings.

Fu et al. [11] introduced a credential ticket-based HO authentication mechanism that is both quick and secure for hybrid networks that use WiMAX and Wi-Fi. The suggested technique drastically shortens the HO authentication delay by having the Mobile station (MS) and board base stations (BS)/AP finish the validation process and generate their associated public key without engaging with the Identification, Validation, and Accountability server. The formal verification using the AVISPA tool demonstrates the security of the proposed method against multiple malicious assaults, and the strategy satisfies the main security procedures in HO identification semantics. To effectively identify assaults against wireless networks, Nivaashini and Thangaraj [12] proposed ML-based (WIDS), and an ML model has been employed to classify Wi-Fi network records as either standard or one of the specific assault classifications. When the qualities are more discriminatory and delegated, an IDS functions much better. Various attribute selection strategies have been explored to zero in on the most pertinent characteristics of the WIDS; however, fast IDS is a must.

To enhance the IDS performance considerably, Usha and Kavitha [13] suggested a normalized gain-based IDS for MAC Intrusions (NMI). OFSNP and DCMI are the two main pillars of the proposed NMI. The first is a semi-supervised clustering technique called optimum selection using NG and PSO (OFSNP). In contrast, the second is an SVM (support vector machine) classifier for detecting and categorizing MAC 802.11 intrusions (SSC). The suggested NMI finds a better balance between detection performance and training time thanks to the SSC, which is based on PSO (particle swarm optimization) and uses unlabelled performance data to determine a collection of optimum features.

The "EvilScout" system, developed by Shrivastava et al. [14] and based on the LAP's knowledge of how IP prefixes are assigned, is designed to discover and counteract evil twins. EvilScout takes advantage of the SDN's probable for detecting an evil twin deprived of requiring any new hardware or modifications on the AP or client side. Furthermore, the data accessible at the SDN controller paves the way for easier and more precise evil twin identification. This study demonstrated the Effective identification of the evil twin with precision and low operational costs at the SDN Wi-Fi, which also provided the deployment of EvilScout over an open SDN Wi-Fi test platform with a genuine evil twin. Real BSs need not be jammed for the attack to proceed undetected by other users.

The BS tags and other WiFi-hotspot tags were the basis for a unique defensive mechanism described by Ye et al. [15]. Here, demonstrate a concealment attack model based on a replay attack in MSNS (mobile social network service) and then use the unpredictable and reproducible nature of BS tags to verify the spatial-temporal properties of geolocation. In addition, a tag verification technique that utilizes fuzzy extractors was developed, which can successfully accommodate the better bit error rates of Wi-Fi transmission. The bloom filter was established to compress portions of actual hotspot frames while ensuring high randomness. Nevertheless, this setup can be tricked by an attacker pretending to be in a different place.

Sethuraman et al. [16] suggested a powerful WIDS that can identify both abnormalities and intrusions to identify wireless attacks. Integrating a KDE (Kernel density estimation) with a hidden Markov model (HMM) via a TQF (tandem queue with feedback) is used to represent WIDS and improve detection accuracy during active-mode attacks. The suggested KDE-HMM technique/method produces successful outcomes by combining probabilistic and statistical benefits. But in passive mode, deprived of connectivity to the AP, it is difficult to spot the attack unless there is previous knowledge of attack patterns.

Deduction: Newer technologies like the IoT (Internet of Things) and Cyber-Physical Systems have recently contributed to Wi-Fi's adoption. Wi-Fi also facilitates quick setup and connectivity, but many networks are not safe because they are not password-protected or because users freely share their passwords. Furthermore, hackers know the current security measures and are developing novel methods to hack the destination. Drones are used in wireless attacks that target wirelessly as ML usage skyrockets. Each of these methods has its benefit that mitigates some of the others' shortcomings. There remains an assortment of problems that need to be addressed. Accordingly, a robust security system that makes the routine of cutting-edge DL techniques is required.

Proposed Methodology

Fig. 1 is a schematic diagram of the various implemented training techniques. Considered is a CDBN classifier, optimized by a GA of the scheme's lower branch and given a training set resulting from the AWID dataset. This classifier will be referred to as ODBN. By taking into account collective statistics derived from such MAC frames that are comprised of adequately selected time windows, this OCDBN applies it to a novel dataset created from the pre-processed AWID dataset. The strategy is geared toward post-transmission analysis. Algorithmically generated rules that take into account updated acquired data. Quite intriguingly, the outcomes demonstrate the Algorithm's efficacy from both the identification and features interpretation perspectives, presenting an indication of the wealth of information that may use to detect suspicious frames.

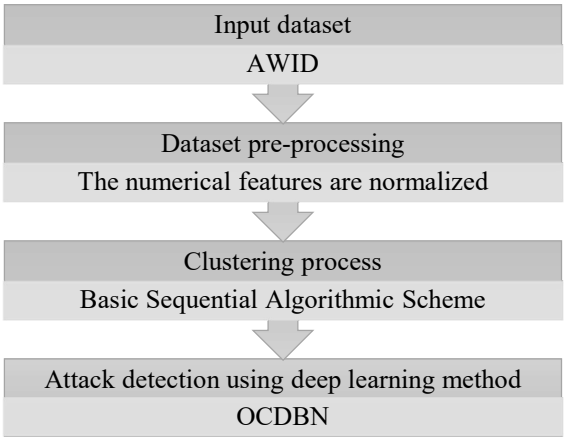

Fig.1. The general framework diagram of the proposed methodology

Dataset description: The AWID dataset was developed by recording Wi-Fi data from a SOHO network consisting of 10e different client devices (smartphones, notebooks, smart TVs). Passive network traffic has been logged using a monitor interface. One more specialized machine plays the function of the hostile network. Browsing the web, data transfer, and multimedia broadcasting are only examples of the popular uses that create regular traffic (i.e., traffic that is not impacted according to any attack). An attack notebook running Kali Linux has been used to develop malicious traffic. Modern control MDK3 and other ad-hoc technologies [17] have been used to carry out the assaults. Fourteen distinct classes of attacks have been spawned. The dataset utilized in this work is split into three pieces, each contributing to the definition of the training set S'_{train}.

For instance, the dataset includes a single network trace with samples of all attack types. This research uses two test sets: S_{test1}, which consists of twelve traffic trace files having only genuine ('normal') traffic, and S_{test2}, which consists of 12 traffic trace files comprising both normal and attack traffic. These are the categories of attacks that can expect: Attackers will send out fake **beacon** frames that claim to come from an AP that doesn't exist and announce an arbitrary Extended Service Set Identifier (ESSID). A field indicating how long the transmission will be congested included in both the Clear To Send (**CTS**) and the Request To Send (**RTS**) frames. These attacks— disassociation, de-authentication, and amok—inhibit the messages passing between a station and an AP during the finding and association phases.

An assault takes advantage of empty data frames, and the platform notifies the AP that it will be sleeping by setting a **power-saving** frame signal in those frames. The AP buffer frame is destined for

a sleeping station to prevent unnecessary transmissions. To notify stations that it has to await MAC data frames from them, access points (APs) establish the TIM (Traffic Indication Map) property and broadcast their MAC addresses whenever they issue a beacon. Entails flooding the AP with a large number of **probe demand** messages. The AP must send a probe response to every probe demand it receives, as specified by IEEE 802.11. In **evil twin**, the attacker causes the victim station to connect with a recognized AP by sending notifications with Straightforward Service Set IDs admitted to the victim position. The **cafe latte and** attacks take advantage of the new station's optional ARP message and the beacon and probing response frames to trick neighboring stations into thinking an accessible (established) AP is not.

Forcing a victim station to provide you frames requires sending **ARP** request signals to it. Given the transient nature of the ARP request, an attacker is likely to repeatedly send them to a single victim station To gather sufficient keystream data for additional attacks. In the **chop-chop attack**, the attacker modifies one or more bytes of a legal frame acquired from a victim position and then uses the access point (AP) as an oracle to attempt to predict the equivalent byte(s) of the plaintext. Chop-chop is a **fragmentation** attack that extracts plaintext and smart contract one byte at a time. The fragmentation attack takes advantage of numerous aspects of IEEE 802.11 networks to expedite plaintext recovery.

Proposed OCDBN-based attack detection

After some basic concepts and notation, this section will describe the data science method that was ultimately chosen. Furthermore, in the following areas, explain what each block entails. First, let's use this notation: In addition to the x pattern of features that come with a MAC frame and the n number of features, there have the Lab set of tags, connected with attack classes, and the $NLab$ amount of labels, where $\mathfrak{l}_i$the label of the set Lab, $i = 1,\ldots,NLab$. The AWID dataset has been pre-processed so that the S'_{train} training data set may be utilized for classifier model generation. The S train training data is a compressed set of arranged pairs $(x, \mathfrak{l})$, where x is a pattern and $\mathfrak{l} \in Lab$ is the ground truth tag related to x. $N_{\mathfrak{p}}$ is the cardinality of the training data set in its compressed form, i.e., the number of patterns in S_{train}, and $N'_{\mathfrak{p}}$ is the cardinality of the training set in its uncompressed format, i.e., the number of practices in S'_{train} and also X, the set of patterns from the compressed training set S_{train}; X', the set of patterns from the AWID training set S'_{train}; let $X' = \left\{x_1,\ldots,x_{N'_{\mathfrak{p}}}\right\}$; and, the dispersion parameter of BSAS. Here, focus on patterns that combine boolean, discrete nominal, and numeric either integer or real-valued characteristics. It is expected that a numeric feature is normalized to be within the interval [0, 1]. To determine how different two objects are, this step uses the following definition of $\mathfrak{d}_r$in this study. Assuming r is between 0 and 1, this step has the following expression for normalized numerical features:

$$\mathfrak{d}_r\big(x(r), y(r)\big) = \mathfrak{l} \cdot |x(r) - y(r)| \quad (1)$$

For nominal/boolean features, let:

$$\mathfrak{d}_r\big(x(r), y(r)\big) = \begin{cases} 1 & x(r) \neq y(r) \\ 0 & x(r) = y(r) \end{cases} \quad (2)$$

Finally, describe the overall dissimilarity amount as the average of dissimilarity standards attained for all features:

$$\mathfrak{d}(x,y) = \frac{1}{n}\sum_{r=1}^{n}\mathfrak{d}_r\big(x(r),y(r)\big) \quad (3)$$

where n is the number of measured features, and reliably with the description assumed above, the dissimilarity amount takes principles in the interval [0, 1]. The representation view of the adopted OCDBN training systems is demonstrated in Fig. 2. Assumed the training set mined from the AWID dataset, considered CDBN with GA.

Data Pre-processing and clustering: The 156 attributes are whittled down to a manageable number, and the numerical features are normalized by hand. The goal is to specify a small enough yet sufficiently comprehensive data collection to improve the classification performance. The resulting S'_{train}, is the product of this stage, BASA clustering technique is used to minimize the cardinality of the initial training set and produce a large number of little clusters, every of which is labeled with its characteristic pattern. The key is to minimize the primary key of the pattern set in a way that is sensitive to the information contained inside it. One can adjust the amount of information you lose or gain by tweaking the threshold in this information granulation technique depending on a dissimilarity metric. The final product of this process is the compressed S_{train}. There are three different types of features: numeric, nominal, and boolean. The maximum value for each numerical characteristic is determined by consulting the IEEE 802.11 specifications [18], and the range for all numerical features is from 0 to that maximum value. Then, normalize each characteristic f as $fn = \frac{f}{f_{max}}$.

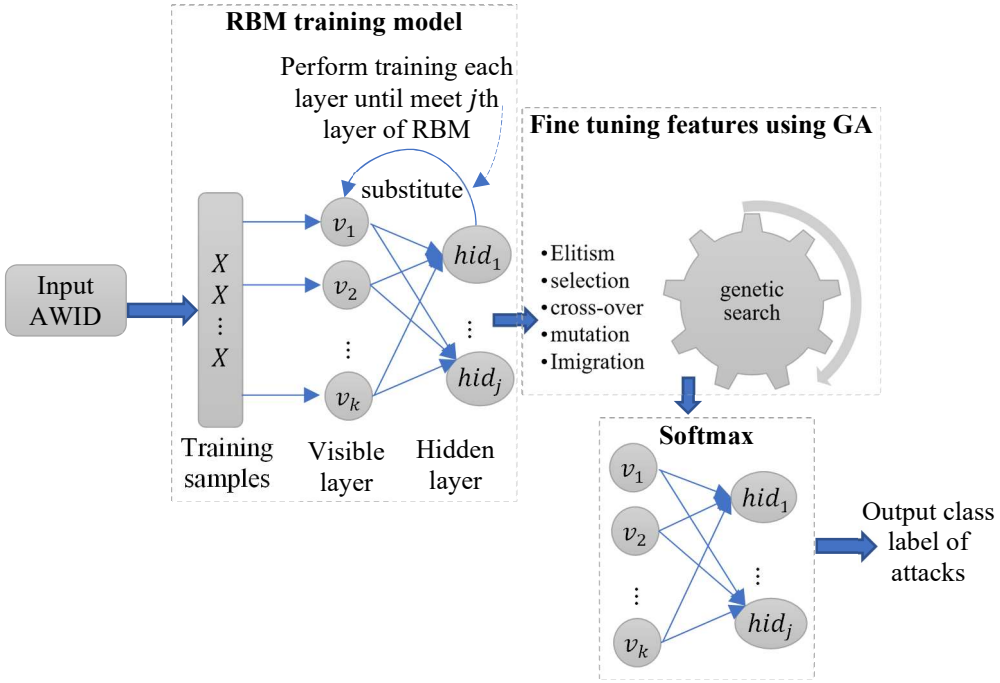

Fig.2. The schematic view of the adopted OCDBN training systems for attack detection

CDBN: In the DBN [22], restricted Boltzmann machines (RBM) are used to extract deep features from the input dataset and are layered one on top of the other in a multilayer configuration. Equation (4) depicts the joint perspective distribution between input data x and the $hidl$-layer hidden layer $hidl^k$in the observable layer. To determine the significance of the data, the unsupervised greedy technique is used. It is customary to begin RBM learning by establishing the base layer's model parameters. After

that, the output of the $hidl$ of the initial RBM layer is utilized as the input for the RBM of the second layer, and the parameters of the primary layer are gradually learned. As the last hidden layer, the softmax regression classifier is paired up, and the automated gradient descent technique is used to fine-tune the model (Almanaseer et al., 2021; Dai et al., 2020).

$\mathrm{P}(x_m, hidl^k) = \left(\sum_{k=0}^{l-2} P(hidl^k \mid hidl^{k+1})\right) * \mathrm{P}(hidl^{l-1}, hidl^l)$	(4)

The probability distribution $\mathrm{P}(hidl^{l-1}, hidl^l)$ between the topmost RBM's visible and hidden layers have been defined. Following the initialization phase, a set $x_n$of training, feature maps are provided. After initialization, the jth hidden layer has no effect on the kth visible layer (vl) in the RBM network topology. Each set is then established before the training time and learning rate are provided. Adjustments to the instruction parameters are made using the comparable distribution technique. If it works, the output will keep going; if not, the parameter training will keep going according to the equation (5-6). As soon as the RBM has been initialized, the first layer of training can begin. The primary RBM layer is trained using the $hidl$ as input, and this process is repeated up to the final RBM layer. The productivity of this layer is then coupled to a fine-tuned softmax model as a classification classifier.

$w = w + \mathrm{I}(hidl_1 x_1' - V(hidl_2 = 1 \mid x_2) x_2')$	(5)
$bs = bs + \mathrm{I}(x_1 - x_2)$	(6)
$vlbs = vlbs + \mathrm{I}(hidl_2 = 1 \mid x_2)$	(7)

Where ε represents the learning rate, V represents the vector $V(hl_2 = 1|x_2)$, w represents the weight element in the filter's row and column, bs represents the bias of each hidden group, while $vlbs$ represents the common bias among all visible units. CDBNs frequently employ Convolutional Restricted Boltzmann machines (CRBMs). CNN's use filters not to discover new information about an object but to construct links between the layers. In the CNN, neurons are only partially connected, while in the DBN architecture, the visible layer of neurons is linked to the hidden layer of neurons. There are no ties between the shown nodes. Likewise, in the graphical architecture of a basic RBM, all hidden nodes have undirected links with one another and are linked to each other. Feature extractors like CDBN, which can generate hierarchical feature architectures, have been becoming more common in recent years for use in pattern recognition. Effective probabilistic inferences can be made both vertically and horizontally with the help of a CDBN model. Many layers of max-pooling CRBMs top the structure, and its training is accomplished in a manner analogous to that of a conventional deep neural network (DBN): via the greedy layer-wise approach. With the help of a Convolutional Neural Network (CDN), the system can learn complex features, such as stroke groups or object components. Two CRBM layers were employed for training the CDBN in the context of the system's tests, and a feed-forward approach was used for inference. Atop CRBM sits the CDBN. A sequence of CRBMs, each of which feeds into the next, is used to train the CDBN method. Sets of localized and common parameters of the CRBM architecture connect the visible and hidden levels. Both real and binary-

valued units are displayed, with the latter being hidden from view [20]. In addition to the three max-pooling layers and the three convolutional layers, this research also makes use of three other kinds of layers. The pre-processed attribute vectors displayed in the preceding frame are fed into the CDBN algorithm via the first layer's output. The inputs are evaluated iteratively, for each iteration consisting of matrix multiplication by the weighted matrix w, with the addition of a bias of bs. The likelihood of features is the ultimate output of CDBN, and these features are labeled for use in attack prediction. Results from incorporating CDBN into a genetic optimization process, which improves accuracy. BSAS supplies a set of representative patterns, $N_p$representatives, which are used as the basis for each classifier. To reduce complexity, GAs only use those features to identify representative patterns for each attack type.

Genetic Algorithm for feature selection: The $n = 25$ features used to characterize patterns in the training set in a GA are being used to choose features for the final model. It is impossible to conduct a complete search for the subset of features best connected to a given categorization challenge. Implementing meta-heuristic optimization approaches, such as GA, is an efficient way to deal with an automatic feature selection challenge. Because when an objective function cannot be expressed in closed form, genetic optimization techniques can be used as a useful alternative. To solve an optimization tricky, a GA must generate a large number of potential solutions over time. For feature selection issues like the one we're solving, genetic code, a size n vector of binary variables, represents each result in the permissible domain. Each aspect of the pattern is examined during training only if the ith element of the genetic code is 1, otherwise, it is ignored. The goal is to optimize some kind of fitness function, which may be thought of as the efficiency with which a classifier uses the features it has learned to prefer through natural selection. This is how the actual implementation functions: Let's call the total number of unique DNA sequences in the population K. The initial population is formed arbitrarily, with a 50% chance that each gene will be set to 1 and the other 50% chance that it will be set to 0. Each person is given a fitness score, and the whole population is ranked from best to worst. Using these guidelines, a new generation is built at each stage.

- Formation of an elite. Only the first $\lceil \varepsilon K \rceil$ people on the ε ranked list of the preliminary population as features are passed on to the following generation.
- The following criteria are used to pick several people from the $K - \lceil \varepsilon K \rceil$ people who are left after elitism has taken its toll. Scanning the ordered list of people who are still around after the elitism is done and picking one at random with probability P is done if si (quantity of selected individuals) = 0.
- The residual sorted list of unselected persons is worth looking at if si is unusual. Iteratively scan the list, picking an item at random with probability P, and stopping when just one person has been chosen and the value of si is increased by one.
- At this stage, this step has chosen si sorted individuals, where si is even and $2 \leq si \leq K - \lceil \varepsilon K \rceil$.
- To implement the **cross-over** procedure, this step will pair up the same persons this step chose in the previous phase. Crossover is the process through which genes from two different people are swapped arbitrarily. From two well-chosen parents, two children are born. Both of the

offspring receive random, unrelated copies of their parents' genetic code. Therefore, a child's genetic code is a composite of those of both parents. In total, there are $\frac{si}{2}$ crossovers, each of which gives rise to si distinct members of the next generation.

- Each of the si offspring produced by the cross is randomly selected for **mutation** with a probability of mP. When a mutation occurs, a gene is flipped randomly with probability 0 independently.
- The remaining $K - \lceil \varepsilon K \rceil + si$ people of the upcoming generation are defined arbitrarily by assigning a value of 1 to each gene with a probability of 0.5, as part of the **immigration** process. After a certain point, the evolutionary optimization process is halted since there is no longer much room for the fittest individual to get much better. It is recommended that K = 10, ε= 0.1, P = 0.8, and mP = 0.3 be used in the implementation. As a bonus, GA can be utilized to zero down on the best traits for each unique type of attack. Finally, attack-specific characteristics are used to categorize each attack type.

Experimental results and discussion

The ability to evaluate how well the suggested OCDBN defines dissimilarity was made possible by this method. A GA technique can be wrapped around it to pick traits that are relevant to each attack. The structural complexity of classification models can be reduced and the key information needed to recognize each assault can be better understood with the help of feature selection. Analyze the mean squared error (MSE) for each of the 12 files that make up S_{test} (S_{test2}), where N is the entire number of classifications, b_i is the number of misclassified patterns in class i and r_i is the total number of patterns in class i. Have $N = NLab$ for the multi-class problem, whereas for the one-class-against-all strategy, N=2. MSE and scattering index (SI) is employed as performance metrics for the proposed model to further assess the proposed method's adaptability. Using the followingly constructed identification accuracy and false positive rate (FPR) metrics, this step may verify the classification accurateness of the suggested method of wrapper optimization with DL classification.

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(b_i - r_i)^2} \times 100\% \quad (7)$$

$$SI = \frac{MSE}{\overline{obf}} \quad (8)$$

where N is the number of attributes nominated of features; b_i and r_i are observed and forecast attacks correspondingly; $\overline{obf}$ indicated as average experimental failures established information.

$$\text{Accuracy} = \frac{\text{number of attacks identified}}{\text{total number of attack present}}$$

where A is detection accuracy that is simplified as the ratio of the number of attacks identified to the total number of attacks.

$$FPR = \frac{number\ of\ non\ attack\ data\ identified\ as\ attack}{total\ number\ of\ attacks} \quad (9)$$

Here FPR is classified as the ratio of some non-attack data identified as faulty to the total number of attack data. Table 1 depicts the comparison of statistical measures of the proposed method- with the other methods for Wi-Fi networks such as KDE-HMM [16], SVM [13], and Min-Max neuro-fuzzy classifier [21].

Table 1: The comparison of numerical measures of the projected method- with the other methods

Metrics	KDE-HMM	SVM	Min–Max neuro-fuzzy classifier	OCDBN
MSE	3.3529	3.2357	3.0159	2.1841
SI	0.3854	0.3349	0.2758	0.1841
Accuracy (%)	91	92	93	95
FPR (%)	32.4	31	28.9	26

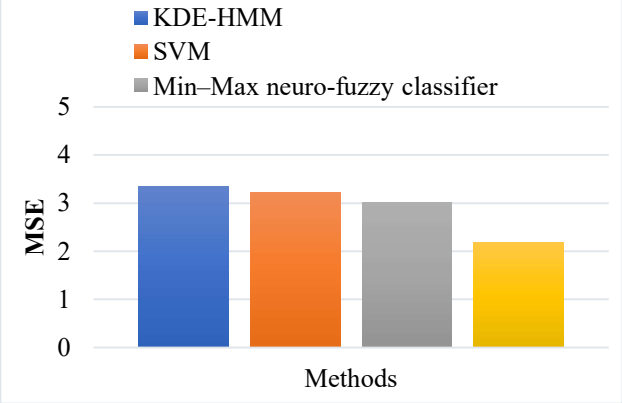

Fig.3. MSE comparison results

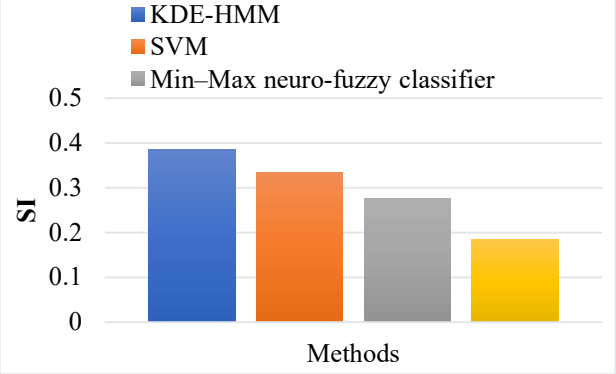

Fig.4. SI comparison results

Fig.3 displays the gain MSE score for the OCDBN classifier, which is about 2.1841. This score is offset and stuck at attack. At a 40% fault rate, the OCDBN classifier's performance degraded because the number of defects was greater and the distance from the margin was more ambiguous. Although

the spike fault is difficult to identify for all classifiers, the percentage of MSE reduced with a high fault rate. FIGURE 4: A comparison of the SI scores across all classifiers. SI evaluates the methods according to how well they can identify network attacks. According to the findings of SI, the OCDBN classifier performed worse than other methods.

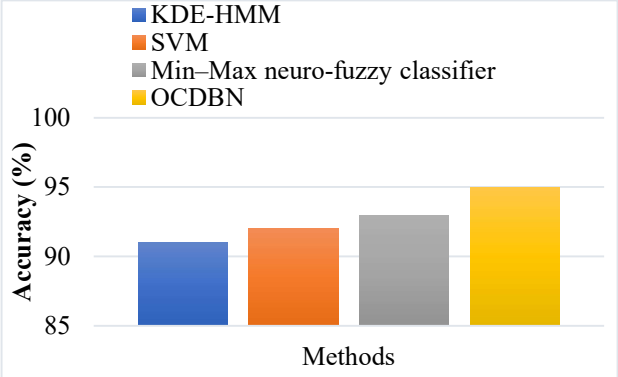

Fig.5. Accuracy comparison results

Figure 5 displays a comparison of the accuracy of various attack-type classifiers. As can be shown in the table, OCDBN outperforms KDE-HMM, SVM, and the Min-Max neuro-fuzzy classifier in recognizing several errors at once. Even with a lot of errors, OCDBN was still able to detect 95% of them. In terms of accuracy, the Min-Max neuro-fuzzy classifier struggles when presented with a sizable dataset. There is a general trend toward decreased accuracy across classifiers when the number of defects rises. Despite the high error rate, OCDBN remains the most reliable method. The OCDBN DL-based technique had the highest bottom detection average accuracy in the experimental evaluation, while yet maintaining a low False alarm rate. CDBN's main strength is that it can learn information adaptively through several gates, making it a more potent instrument for attack diagnosis. The primary takeaway from this scenario is that CDBN was still able to effectively categorize and isolate defects even when many faults were injected at once.

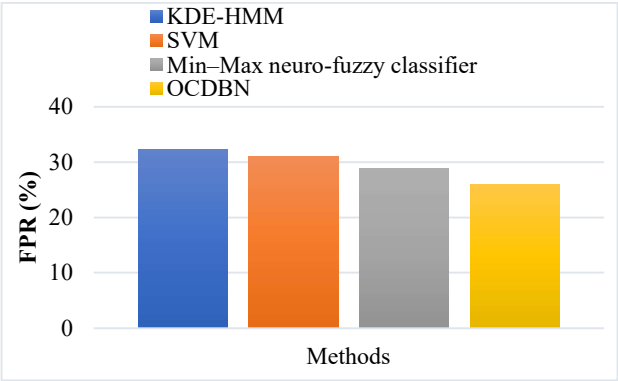

Fig.6. FPR comparison results

When it comes to the false positive rate of the training set, the experimental results of the suggested scheme are compared to those of the KDE-HMM, SVM, and Min-Max neuro-fuzzy classifier. Figure 6 displays a contrast between the suggested method and the baseline approaches based on the FPR metric. The proposed method provides the best FPR value, which contributes to a substantial and

essential improvement of FPR compared to alternatives. Increases range from 26% for SVM to 31% for Min-Max neuro-fuzzy classifier, and 28.9% for SVM to KDE-HMM.

Conclusion and future work

This work planned an ascendable and modular architecture for detecting malicious Wi-Fi traffic using ML techniques. An array of classifiers, each trained to recognize frames from a particular type of attack, forms the basis of the suggested architecture. Since modern wireless networks operate at lightning speeds and are only expected to increase in that regard in the future, this research focuses on determining how feasible it would be to identify fraudulent Wi-Fi traffic at a reasonable cost automatically. Therefore, this study decides to investigate how well attacks may be identified using single-frame classification. In reality, this work can describe a classification model architecture that accommodates the situation to hardware execution at a low cost, making it possible to integrate with low-priced Wi-Fi base stations and access routers. An array of characteristics is used to represent a MAC frame, and a dissimilarity measure is constructed to make comparisons between them. Classification models have been synthesized using OCDBN with the aid of a GA-based per-class feature selection method. Based on their results, the presented algorithms prove that a single-frame classification strategy is doable. Over 95% accuracy is achieved for all 14 attack classes defined in the AWID dataset, proving that malicious frames can be reliably identified. In future work, other swarm-based methods for feature selection can be utilized, in addition, the resource sharing in wifi-networks can be focused.

References

1. Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., & Kambhampati, S. (2020). A survey of moving target defenses for network security. *IEEE Communications Surveys & Tutorials*, *22*(3), 1909-1941.
2. A. Reyes, A., D. Vaca, F., Castro Aguayo, G. A., Niyaz, Q., & Devabhaktuni, V. (2020). A machine learning based two-stage Wi-Fi network intrusion detection system. *Electronics*, *9*(10), 1689.
3. Zhao, Y. Z., Miao, C., Ma, M., Zhang, J. B., & Leung, C. (2012). A survey and projection on medium access control protocols for wireless sensor networks. *ACM Computing Surveys (csuR)*, *45*(1), 1-37.
4. Choi, W., Lim, H., & Sabharwal, A. (2015). Power-controlled medium access control protocol for full-duplex Wi-Fi networks. *IEEE Transactions on Wireless Communications*, *14*(7), 3601-3613.
5. Granato, G., Martino, A., Baldini, L., & Rizzi, A. (2022). Intrusion Detection in Wi-Fi Networks by Modular and Optimized Ensemble of Classifiers: An Extended Analysis. *SN Computer Science*, *3*(4), 1-17.
6. Aminanto, M. E., & Kim, K. (2017). Detecting active attacks in Wi-Fi network by semi-supervised deep learning. In *Conference on Information Security and Cryptography*.
7. Zhu, X., Ding, B., Li, W., Gu, L., & Yang, Y. (2018). On development of security monitoring system via wireless sensing network. *EURASIP Journal on Wireless Communications and Networking*, *2018*(1), 1-10.

8. Dwivedi, S., Vardhan, M., & Tripathi, S. (2021). Multi-parallel adaptive grasshopper optimization technique for detecting anonymous attacks in wireless networks. *Wireless Personal Communications*, *119*(3), 2787-2816.
9. Agarwal, M., Pasumarthi, D., Biswas, S., & Nandi, S. (2016). Machine learning approach for detection of flooding DoS attacks in 802.11 networks and attacker localization. *International Journal of Machine Learning and Cybernetics*, *7*(6), 1035-1051.
10. Mahini, H., & Mousavirad, S. M. (2020). Wi-Fi intrusion detection and prevention systems analyzing: a game theoretical perspective. *International Journal of Wireless Information Networks*, *27*(1), 77-88.
11. Fu, A., Zhang, G., Zhu, Z., & Zhang, Y. (2014). Fast and secure handover authentication scheme based on ticket for WiMAX and Wi-Fi heterogeneous networks. *Wireless personal communications*, *79*(2), 1277-1299.
12. Nivaashini, M., & Thangaraj, P. (2021). Computational intelligence techniques for automatic detection of Wi-Fi attacks in wireless IoT networks. *Wireless Networks*, *27*(4), 2761-2784.
13. Usha, M., & Kavitha, P. J. W. N. (2017). Anomaly based intrusion detection for 802.11 networks with optimal features using SVM classifier. *Wireless Networks*, *23*(8), 2431-2446.
14. Shrivastava, P., Jamal, M. S., & Kataoka, K. (2020). EvilScout: Detection and mitigation of evil twin attack in SDN enabled Wi-Fi. *IEEE Transactions on Network and Service Management*, *17*(1), 89-102.
15. Ye, A., Li, Q., Zhang, Q., & Cheng, B. (2020). Detection of spoofing attacks in WLAN-based positioning systems using Wi-Fi hotspot tags. *IEEE Access*, *8*, 39768-39780.
16. Sethuraman, S. C., Dhamodaran, S., & Vijayakumar, V. (2019). Intrusion detection system for detecting wireless attacks in IEEE 802.11 networks. *IET networks*, *8*(4), 219-232.
17. Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2015). Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. *IEEE Communications Surveys & Tutorials*, *18*(1), 184-208.
18. IS Association. (2012). IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std*, 802-11.
19. Hua, Y., Guo, J., & Zhao, H. (2015, January). Deep belief networks and deep learning. In *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things* (pp. 1-4). IEEE.
20. Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2011). Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, *54*(10), 95-103.
21. Rizzi, A., Granato, G., & Baiocchi, A. (2020). Frame-by-frame Wi-Fi attack detection algorithm with scalable and modular machine-learning design. *Applied Soft Computing*, *91*, 106188.