ISSN 2723-6471

ARP Spoofing Attack Detection Model in IoT Networks Using Machine Learning: Complexity vs. Accuracy

Adeeb Alsaaidah[1,*], Omar Almomani[2], Ahmad Adel Abu-Shareha[3], Mosleh M. Abualhaj[4], Anusha Achuthan[5]

[1,2,4]*Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman, Jordan*

[3]*Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Amman, Jordan*

[5]*School of Computer Science, Universiti Science Malaysia, Penang, Malaysia*

(Received: August 14, 2024; Revised: September 21, 2024; Accepted: September 25, 2024; Available online: October 20, 2024)

Abstract

Spoofing attacks targeting the address resolution protocol, or the so-called ARP, are common cyber-attacks in IoT environments. In such an attack, the attacker sends a fake message over a local area network to spoof the users and interfere with the communication transferred from and into these users. As such, to detect such attacks, there is a need to check the network gateways and routers continuously to capture and analyze the transmitted traffic. However, there are three major problems with such traffic data: 1) there are substantial irrelevant data to the ARP attacks, 2) there are massive patterns in the way by which the spoof can be implemented, the massive patterns describe abnormal or strange patterns in network traffic that can appear as a result of widespread ARP spoofing attacks, and 3) there is a need for fast processing of such data to reduce any delay resulting from the processing stage. Accordingly, this paper proposes a detection approach using supervised machine learning algorithms. The focus of this paper is to show the tradeoff between speed and accuracy to offer various solutions based on the demanded quality. Various algorithms were tested to find a solution that balanced time requirements and accuracy. As such, the results using all features and with various feature selection techniques were reported. Besides, the results using simple classifiers and ensemble learning algorithms were also reported. The proposed approach is evaluated on an IoT network intrusion dataset (IoTID20) collected from different IoT devices. The results showed that the highest accuracy is obtained using the RF classifier with a subset of features produced by the wrapper technique. In such a case, the accuracy obtained was 99.74%, with running time equal to 305 milliseconds. However, if time is more critical for a given application, then DT can be used with the whole feature set. In such a case, the accuracy was 99.41%, with running time equal to 11 milliseconds.

Keywords: Feature Selection, Machine Learning, Machine Learning, Address Resolution Protocol Spoofing Attack, Internet of Things

1. Introduction

The Internet of Things (IoT) is a network of interconnected devices that exchange data via recognized protocols [1], [2]. IoT networks have been distributed worldwide to aid the development of smart cities, smart agriculture, and smart transportation [3], [4], [5], [6]. The amount of data transmitted, processed, and collected through such networks is enormous. The benefits of such data are manifold, including enhanced decision-making, improved operational efficiency, and deep insights into user behaviors and system performance. Such data also enables the development of advanced analytics and machine learning models that can solve various problems and facilitate personalized services, better customer experiences, and effective resource management. Besides, data in some sectors, like healthcare, can save lives, improve safety, and foster healthcare services. As such, IoT solves various human life-related problems and challenges.

The widespread use of IoT networks and devices increases the number of cybersecurity attacks [7], [8]. One of the most common and critical attacks that affects the IoT network is the Man-in-the-middle (MITM) attack. MITM attacks frequently incorporate Address Resolution Protocol (ARP) spoofing, allowing hackers to intercept and alter communications between IoT devices and connected servers [9]. Since ARP spoofing can lead to network traffic

***Corresponding author: Adeeb Alsaaidah (a.alsaaidah@ammanu.edu.jo))**
DOI: https://doi.org/10.47738/jads.v5i4.374

 ISSN 2723-6471

disruption, manipulation, and eavesdropping, there is a significant risk to the security of IoT networks and devices [10]. In IoT systems, ARP spoofing is more prevalent and can have tragic results. Because of the subsequent factors: lack of security cooperation, host unpredictability, and open media access [11]. Defeating ARP spoofing attacks in an IoT environment can be implemented using ARP inspection, intrusion detection system (IDS), intrusion prevention systems (IPS), authentication, and encryption. However, IoT devices cannot operate such solutions due to their limited processing and storage capacity [12].

However, the IDS can be operated based on trained models to save the space and processing time of the operated devices, which enables the detection of any anomalies. However, to allow the IDS to recognize the threat, a model for the IDS must be trained with instances of the targeted attack, side by side, with instances of normal traffic [13], [14]. Using such mechanism, various machine learning models have been used to create the trained model and used the training data to recognize the type of unseen traffic, such as support vector machine (SVM), random forest (RF), Extreme Gradient Boosting (XGBoost) [15], [16].

Machine learning algorithms generally require massive training data for highly accurate prediction [17]. Processing such massive data with large dimensionality is challenging and hindered by the low computational power of IoT devices. As such, feature selection reduces the dimensionality while preserving the information to maintain high performance. As such, IDS for IoT places a high value on reducing the complexity of data by eliminating irrelevant features [18]. FS approaches reduce network data size by removing irrelevant data and maintaining the relevant ones. Additionally, the computing burden of IDS is reduced, and the detection speed is improved using feature selection. As a result, feature selection is one of the most crucial elements of data preparation in IDS because it influences the detection's accuracy [19]. To deal with IDS, several feature selection techniques based on metaheuristics have been adopted [20], [21]. However, there is no unique and widely acceptable technique for feature selection in all the domains. Besides, feature reduction often might reduce the time complexity but decrease the performance, which can be considered the tradeoff between the ML techniques' speed and accuracy.

As such, the contributions of this paper can be summarized as follows: 1) Experimenting with various feature selection techniques on the ARP Spoofing attack detection. 2) Use simple and ensemble classifiers instead and compare their prediction accuracy. 3) Evaluating the performance of recent feature selection and classification trends using accuracy, precision, sensitivity, F-measure, and speed. The remainder of the paper is structured as follows. Section 2 presents the related works. The proposed model is explained in Section 3. The findings obtained and experiment setup are discussed in Section 4. Section 5 provides a paper conclusion.

2. Related Work

Feature selection is commonly used with machine learning models, especially when the input datasets are complex and huge. This stage involves selecting the relevant features and discarding the irrelevant ones to speed up and ease the machine-learning algorithm. Feature selection methods are classified into filter, wrapper, and embedded types. Filter methods evaluate the relevance of features by their correspondence with the target variable and select the most correlated ones. Wrapper methods use a machine learning algorithm to evaluate various subsets of features and select the optimal subset by searching through the feature space. Embedded methods select feature features as part of the model training [22]. In addition to these traditional methods, optimization, and meta-heuristic algorithms are used for feature selection due to their ability to search large and complex feature spaces effectively. Meta-heuristic algorithms, such as Genetic Algorithms (GA), Particle Swarm Optimisation (PSA), and Ant Colony Optimization (ACO), are used for such tasks [23].

Recent trends in classification have seen significant advancements with the development of sophisticated algorithms designed to enhance model performance and efficiency. Gradient Boosting (GBoost) and its variants, such as XGBoost, have gained prominence due to their superior capabilities in handling complex classification tasks. These algorithms are ensemble learning techniques that build multiple models and combine weak learners, typically decision trees, to form a robust predictive model.

ISSN 2723-6471

Among various meta-heuristics algorithms, the focus will be on the FireFly Optimization Algorithm (FFO) [24]. Nevertheless, what can be implemented using FFO can also be implemented using other algorithms, such as Grey Wolf Optimization (GWO) and White Whale Optimization (WWO) algorithms. Xu, et al. [25] proposed an improved FFO that combines the binary FFO algorithm with opposition-based learning (OBL) for feature selection in classification tasks. There were two ideas for the proposed improvements: using a binary representation of the FFO instead of the continuous representation and using the OBL to initialize the population. The improved FFO outperforms Particle Swarm Optimization (PSO) and the conventional FFO algorithm for feature selection over 10 classification-based datasets. Several improvements have been proposed for the Firefly Algorithm to enhance its optimization capabilities. These enhancements include random firefly movements to balance exploration and exploitation [26]. Another approach involves integrating the Gravity Search Algorithm (GSA) with FFO to create a hybrid method, which shows faster convergence and better results than the individual methods [27]. An improved Firefly Algorithm was proposed in the search process to solve non-convex problems by incorporating mutation mechanisms, dynamic parameter adjustments, and dynamic tolerance mechanisms [28].

A combination of feature selection and classification algorithms have been used in various IDS-related applications. An approach for FEATURE SELECTION based on FFO for network intrusion detection is proposed by Selvakumar and Muneeswaran [29], which was implemented using the KDD CUP 99 dataset. Based on the conducted experiments, FFO leads to selecting features that allow for detecting intrusions more accurately with only 10 selected features. The classification task uses a simple decision tree classifier and Bayesian Networks (BN). The advantage of the proposed model is proof of the efficiency of the FFO algorithm and the effectiveness of selected 10 features out of 41 in the KDD. The results of the selected features reported accuracies ranging between 44% to 99% compared to accuracies of 26% to 97% for utilizing the whole set of features for U2R, R2L, Probe and DoS attacks.

Almomani [20] proposed a feature selection model for Network IDSs (NIDSs) based on various optimization algorithms such as PSO, GWO, FFA, and GA. The model is evaluated using the UNSW-NB15 dataset, SVM, and J48 ML classifiers. The results showed that the reduced feature set produces better results than the whole set. Besides, the FFO showed a good performance, outperforming the GWO and PSO.

Rajabi, et al. [30] developed a model based on FFO and a fast-learning neural network. The proposed model was evaluated and compared to other models built based on fully connected neural networks and SVM classifiers. Besides, multiple meta-heuristic feature selections, including PSO and GA, were used for comparison. The model is evaluated using the KDD dataset, and the results showed that the fast-learning classifier with FFO outperforms the other approaches.

Almomani [21] proposed a hybrid IDS model combining bio-inspired metaheuristic algorithms to detect various attacks. The model aims to reduce the number of features and improve the classification accuracy. PSO, multiverse optimizer (MVO), GWO, moth-flame optimization (MFO), whale optimization algorithm (WOA), FFA, and bat algorithm (BAT) were used for feature selection. The proposed approach was assessed based on the UNSW-NB15 dataset and using J48, SVM, and RF classifiers. The MVO-BAT model reduces features to 24 features and obtained the same results as those acquired using all features, whereas the MFO-WOA and FFA-GWO models decrease features to 15 with a reasonable accuracy.

Ghosh, et al. [31] proposed a Modified-Firefly Algorithm (MFA) for feature selection with building cloud-based IDS. The model is assessed based on the NSL-KDD dataset. Similarly, Mohammad, et al. [32] proposed a multilayer bio-inspired feature selection model for intrusion detection. The model uses PSO, GWO, and FFO in layer 1 to assign priority values for the undelying features. GA is then used in layer 2 of the model to select a subset of features based on the priority value assigned in layer 1. The model was evaluated based on two well-known datasets, UNSW-NB15 and NSL-KDD, using various evaluation criteria such as precision, recall, and F-Measure. The results demonstrate the powerful and promising mechanism of the proposed model for IDS.

In summary, various meta-heuristic algorithms were used with various simple classifiers for IDS. The results of the reviewed approaches depend on the classifier and the selection method. However, to highlight the performance of these

 ISSN 2723-6471

techniques, a comparison with a simple feature selection approach and ensemble classification is required, which will be conducted in this paper.

3. The Proposed

The proposed model for ARP spoofing detection consists of dataset preparation, preprocessing, feature selection, classification and evaluation, as shown in figure 1.

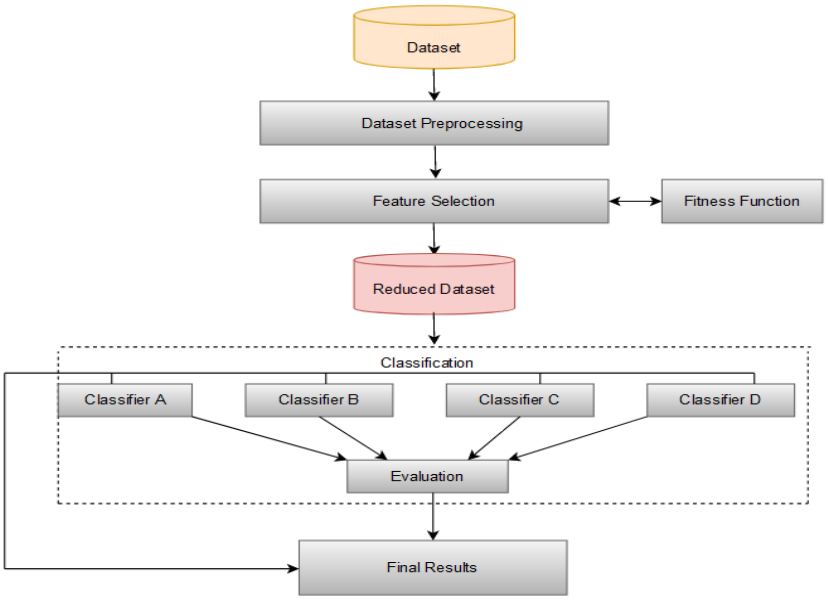

Figure 1. The flowchart of the proposed model

3.1. Dataset Description

The IoTID20 dataset was mainly built to detect cyber attacks in IoT, and it was collected from a smart home's IoT ecosystem, and it is available to the public [33]. Such an ecosystem involves a wireless access point (Wi-Fi), a Wi-Fi router, Wi-Fi cameras (EZVIZ), laptops, smartphones, tablets, AI Speakers (SKTNGU), and other interconnected devices. The other devices in the ecosystem represent the IoT attacking devices, which target the cameras and AI speakers. The Network Mapper (Nmap) tool was used to simulate various real attacks on the IoT ecosystem, as illustrated in figure 2.

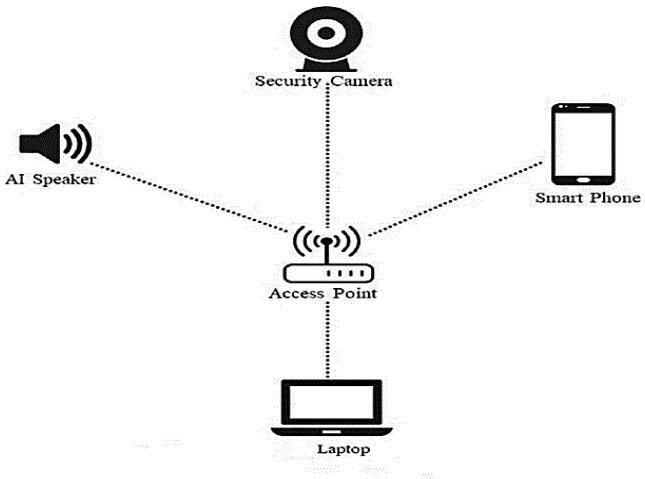

Figure 2. The IoTID20 dataset testbed [34]

The IoTID20 dataset comprises normal traffic as well as four different IoT attack types (DoS, Mirai, MITM, and Scan), as illustrated in figure 3. The IoTID20 dataset has 625,784 instances, 83 features, and 3 Classes (Lable, Cat, sub-cat).

 ISSN 2723-6471

Out of these instances, the developed model focuses on detecting ARP spoofing attacks, which is a type of attack under the MITM category. Therefore, ARP spoofing attacks and normal traffic were collected from the original dataset. Because the IoTID20 dataset contains realistic simulations of ARP spoofing in IoT contexts, it was appropriate for the detection of ARP spoofing.

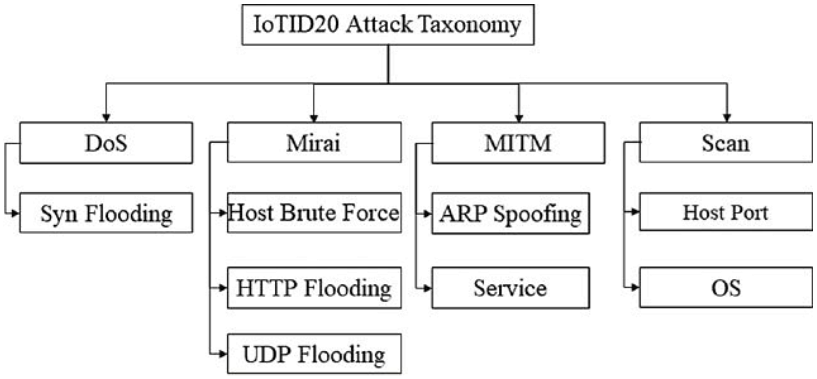

Figure 3. IoTID20 Dataset Attack Taxonomy [35]

3.2. Preprocessing

The preprocessing stage converts the input data into a better form to improve its representation, positively influencing the classification results. This stage consists of several steps: 1) Remove inappropriate features like (Flow_ID, Src_IP, Timestamp, and Dst_IP), which do not contribute to the classification process. 2) Remove the duplicate data to eliminate prediction bias towards these duplicated records. 3) Replace the missing values of the dataset using the median values, the median is a better choice as it is robust to extreme values. 4) Normalize the values using min-max normalization to eliminate bias towards features with a broader range of values.

3.3. Features Selection

Feature selection is a pivotal step in machine learning that involves identifying and utilizing the most relevant features from a dataset. As the feature selection is implemented, a reduced set is generated, which will reduce the classification time and may or may not improve the accuracy of the classification results.

3.4. Fitness Function

The fitness function evaluates the solution in the wrapper and meta-heuristic techniques. Generally, the fitness function is calculated based on the performance of the selected features and their counts. The higher the performance and the less count, the better the solution. In the implemented model, the KNN classifier is utilized because of its simplicity and effectiveness, which allows it to be converted efficiently into a fitness function. The fitness function is evaluated using Equation 1.

$$\text{Fitness function} = \alpha\,\gamma + (1-\alpha)\,\frac{|R|}{|N|} \quad (1)$$

where γ is the KNN classification error rate, |R| is the number of selected features, |N| is the number of features, and α is a weight parameter in the range [0, 1].

3.5. Classifiers

Both simple and ensemble classifiers will be tested in the implemented model. Simple classifiers include algorithms like DT and SVM. These models are straightforward to interpret. For a simple classifier, Naïve Bayesian and SVM will be implemented. KNN and DT will also be implemented, which will consume more time. As for the ensemble classifiers, RF XGBoost, AdaBoost, GradientBoost, CatBoost, and Voting will be implemented.

 ISSN 2723-6471

4. Results and Discussion

All experiments were tested in Python and performed on an i7-1065G7 processor running at 1.50 GHz and 16.0 GB of RAM. Table 1 shows the used classifier parameters setting.

Table 1. classifier parameters setting

Classifier	Setting	
XGB, AdB , GB, and CatB	random_state	1
	learning_rate	0.01
KNN	n_neighbors	5
	weights	uniform
	leaf_size	30
SVM	Kernel	Radial Basis Function
	Regularization Parameter (C)	1
	Gamma	1 (Scale)
DT	criterion	entropy
	splitter	best
	min_samples_split	2
RF	criterion	entropy
	min_samples_split	2
	n_estimators	10
NN	Optimizer	adam
	Layers	3 (64, 32,1)
	Activation function(s)	ReLU, ReLU, Sigmoid

4.1. Evaluation Metrics

Five assessment evaluation metrics were employed: accuracy, precision, sensitivity, F-measure, and execution time. The confusion matrix reports the number of successfully predicted samples (TP and TN) and the number of wrongly predicted samples (FP and FN). Figure 4 illustrates the binary confusion matrix.

		Predicted	
		Normal	Attack
Actual	Normal	TP	FN
	Attack	FP	TN

Figure 4. Confusion Matrix

The percentage of accurately predicted instances (TP and TN) relative to the total number of instances is called accuracy. The accuracy of the mode is calculated using Equation 2.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2)$$

Sensitivity, sometimes referred to as recall, is the percentage of real positives (attacks) that the model properly detects, as given in Equation 3.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

The precision measures how many positive predictions (attacks) are successfully predicted, as calculated using Equation 4.

 ISSN 2723-6471

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

To improve the contradiction between sensitivity and precision, the F1 measure represents the tradeoff between them. The F1 measure is calculated using Equation 5.

$$\text{F} - \text{Measure} = \frac{2\ x\ \text{Precision}\ x\ \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \quad (5)$$

4.2. Feature Selection Outputs

In the proposed model, three feature selection techniques were implemented to enhance the performance of the utilized classifiers. The first technique was a correlation filter-based approach. This method evaluates the correlation between each feature and the target variable, selecting those with high correlation to the target and low inter-correlation with each other. By eliminating redundant and irrelevant features, this technique helps reduce dimensionality and improves model accuracy and interpretability. For this technique, a threshold of 0.8 was selected. All features with correlation below such threshold were discarded. The selected features based on this technique are listed in table 2, with 15 features instead of 79 for detecting ARP Spoofing attacks. This technique algorithm reduced about 81% of the IoTID20 dataset features, thus reducing the execution time.

Table 2. Selected Features based on the Correlation

#	Feature	#	Feature	#	Feature	#	Feature
1	Dst_Port	2	Fwd_PSH_Flags	3	Fwd_URG_Flags	4	Fwd_Byts/b_Avg
5	Fwd_Pkts/b_Avg	6	Fwd_Blk_Rate_Avg	7	Bwd_Byts/b_Avg	8	Bwd_Pkts/b_Avg
9	Bwd_Blk_Rate_Avg	10	Init_Fwd_Win_Byts	11	Fwd_Seg_Size_Min	12	Active_Mean
13	Active_Std	14	Active_Max	15	Active_Min		

The second technique employed was the forward selection wrapper-based approach. This method starts with an empty model and iteratively adds features that improve the model's performance based on a chosen evaluation metric. Considering feature interactions and tailoring the feature selection process to the specific learning algorithm often leads to better model performance than filter methods. The selected features based on this technique are listed in table 3, with 39 features instead of 79 for detecting ARP Spoofing attacks with a 50.6% reduction rate.

Table 3. Selected Features based on the Wrapper Technique

No	Feature	No	Feature	No	Feature	No	Feature
1	Src_Port	2	Dst_Port	3	Protocol	4	TotLen_Fwd_Pkts
5	Fwd_Pkt_Len_Max	6	Fwd_Pkt_Len_Min	7	Fwd_Pkt_Len_Mean	8	Fwd_Pkt_Len_Std
9	Fwd_IAT_Tot	10	Fwd_IAT_Std	11	Fwd_IAT_Max	12	Flow_IAT_Min
13	Fwd_PSH_Flags	14	Fwd_URG_Flags	15	Bwd_URG_Flags	16	Fwd_Pkts/s
17	FIN_Flag_Cnt	18	SYN_Flag_Cnt	19	RST_Flag_Cnt	20	ACK_Flag_Cnt
21	URG_Flag_Cnt	22	CWE_Flag_Count	23	ECE_Flag_Cnt	24	Fwd_Seg_Size_Avg
25	Fwd_Byts/b_Avg	26	Fwd_Pkts/b_Avg	27	Fwd_Blk_Rate_Avg	28	Bwd_Byts/b_Avg
29	Bwd_Pkts/b_Avg	30	Bwd_Blk_Rate_Avg	31	Subflow_Fwd_Byts	32	Init_Fwd_Win_Byts
33	Fwd_Act_Data_Pkts	34	Fwd_Seg_Size_Min	35	Active_Mean	36	Active_Std
37	Active_Max	38	Active_Min	39	Idle_Std		

Finally, we used a firefly meta-heuristics-based technique. Inspired by the behavior of fireflies, this optimization algorithm explores the feature space to find the optimal subset of features. The experiments used a population size of 30 and 100 iterations for meta-heuristics. The selected features based on the proposal model are listed in table 4. The technique reduced the number of features to 23 with a reduction rate of 70.9% of the IoTID20 dataset features.

ISSN 2723-6471

Table 4. Selected Features based on FFO

#	Feature	#	Feature	#	Feature	#	Feature
1	Src_Port	2	Dst_Port	3	Fwd_Pkt_Len_Max	4	Fwd_Pkt_Len_Mean
5	Fwd_Pkt_Len_Std	6	Bwd_Pkt_Len_Max	7	Bwd_Pkt_Len_Mean	8	Flow_IAT_Std
9	Flow_IAT_Min	10	Fwd_IAT_Mean	11	Bwd_IAT_Std	12	Bwd_IAT_Max
13	Fwd_PSH_Flags	14	Bwd_Header_Len	15	SYN_Flag_Cnt	16	ECE_Flag_Cnt
17	Fwd_Byts/b_Avg	18	Fwd_Blk_Rate_Avg	19	Subflow_Bwd_Byts	20	Init_Fwd_Win_Byts
21	Fwd_Act_Data_Pkts	22	Active_Std	23	Idle_Max		

As noted, the filter-based method results in a subset of features that are either included in the subset generated by the wrapper or the FFO algorithms. On the other hand, the FFO and Wrapper algorithms generated subsets that contain intersected various features. However, each includes unique features not present in the other subset, as denoted in the highlighted cells of table 3 and table 4.

4.3. Evaluation of ML Classifier

The experimental evaluation results for each classifier are listed in table 5.

Table 5. Results of the Proposed Model

		XGB	AdB	GB	CatB	Vot	KNN	NB	SVM	DT	RF	NN
Accuracy	All	97.56	97.13	96.52	97.33	97.80	99.00	92.09	97.52	99.41	99.40	97.86
	FFO	99.00	98.55	97.56	98.47	99.17	99.19	92.87	96.86	97.83	99.53	97.82
	Filter	90.78	90.88	91.18	90.84	90.91	82.49	90.32	90.32	91.05	91.27	90.32
	Wrapper	99.34	98.73	99.26	97.93	99.09	99.39	82.85	94.06	98.89	99.74	95.87
Precision	All	96.96	96.65	95.76	97.20	97.27	98.53	87.54	95.94	98.82	99.02	96.40
	FFO	98.21	97.89	96.56	98.22	98.42	98.83	89.87	95.26	95.76	99.20	96.24
	Filter	83.89	84.09	84.23	84.04	84.12	88.68	84.04	84.04	84.21	84.36	84.04
	Wrapper	98.90	97.96	98.57	96.81	98.65	99.19	74.39	90.14	98.05	99.62	93.37
Sensitivity	All	97.90	97.31	96.94	97.14	98.11	99.34	96.93	98.90	99.94	99.72	99.13
	FFO	99.70	99.06	99.47	98.55	99.86	99.45	95.55	98.17	99.78	99.81	99.22
	Filter	99.43	99.34	99.88	99.31	99.35	71.82	97.97	97.97	99.59	99.89	97.97
	Wrapper	99.71	99.37	99.86	98.84	99.42	99.51	96.71	98.07	99.60	99.84	98.15
F-measure	All	97.43	96.98	96.35	97.17	97.69	98.94	91.99	97.40	99.38	99.37	97.75
	FFO	98.95	98.47	97.47	98.39	99.13	99.14	92.63	96.70	97.73	99.51	97.71
	Filter	91.00	91.08	91.39	91.04	91.10	79.37	90.47	90.47	91.26	91.47	90.47
	Wrapper	99.30	98.66	99.21	97.82	99.03	99.35	84.10	93.94	98.82	99.73	95.71
Time	All	67	281	750	21	1150	20201	12	11254	11	325	590
	FFO	51	212	723	26	1096	19542	7	10211	11	296	520
	Filter	42	201	700	15	1011	18542	9	10251	10	285	490
	Wrapper	53	261	752	15	1062	19658	7	10351	11	305	552

 ISSN 2723-6471

The results showed that the RF classifiers outperformed all other classifiers when implemented with the features extracted using the wrapper technique. As noted, the features of the wrapper technique produced the best results of all the classifiers. The RF model always performs the best except for the accuracy using all features in which the DT classifier achieved the best. DT also achieved the best sensitivity and f-measure when used with all features. The results also showed that DT is much faster than RF, even when utilized with the whole feature set, and produces results comparable to those of RF. It was noted that the ensemble algorithms do not overperform RF or DT. Besides, it was noted that the FFO feature selection, although it produced good results, is still not as good as the wrapper technique. A summarise the accuracy and execution time is given in figure 5 and figure 6. In conclusion, the RF model can be used to maintain high accuracy with a subset of features produced by the wrapper technique. If time is more critical for a given application, then DT can be used with the whole feature set.

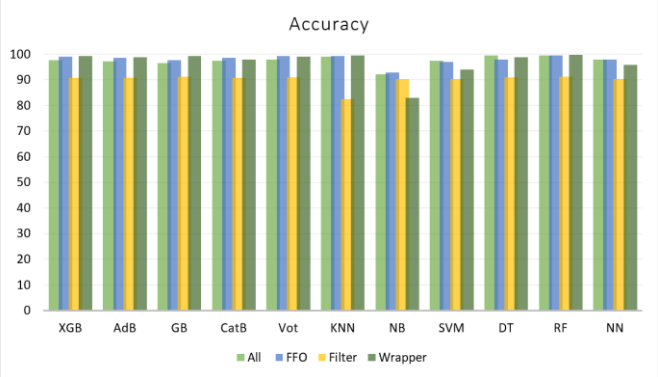

Figure 5. Accuracies of the Classification Algorithms

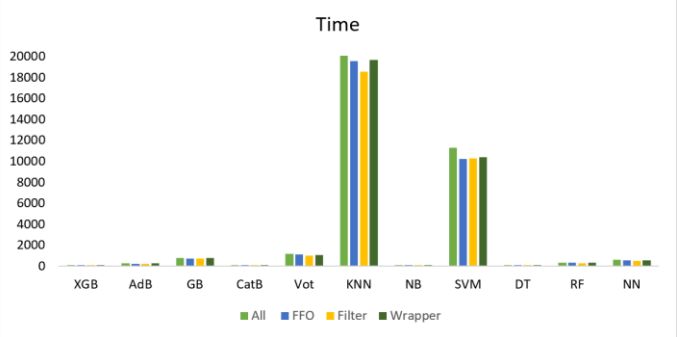

Figure 6. Execution Times of the Classification Algorithms

5. Conclusion and Future Work

This paper proposed a model for detecting ARP Spoofing Attacks in IoT Networks. The proposed model compared the results of multiple classifiers and feature selection techniques. In the dimension reduction stage, the features of the IoTID20 dataset were reduced from 79 to 15, 23, and 39 by the filter, FFO, and wrapper techniques, respectively. In the classification stage, the proposed model applied eleven classifiers of simple and ensemble types. The proposed model was evaluated using accuracy, precision, sensitivity, F-measure, and execution time. The experimental results showed that wrapper feature selection with an RF classifier reduced around 50.6% of the IoTID20 features and achieved an accuracy of 99.74% with running time equal to 305 milliseconds. A limitation of the paper is that it only focuses on reducing the features using filter, FFO, and wrapper techniques and classifying the ARP attack based only on the IoTID20 dataset. In future research, a new Dataset for IoT with different types of attacks will be considered, as well as different metaheuristic algorithms will be considered for feature selection.

6. Declarations

6.1. Author Contributions

Conceptualization: A.A., O.A., A.A.A., M.M.A., and A.A.; Methodology: O.A., M.M.A., and A.A.A.; Software: A.A.A. and A.A.; Validation: O.A., A.A.A., and A.A.; Formal Analysis: O.A., A.A.A., and A.A.; Investigation: A.A.A. and M.M.A.; Resources: A.A.; Data Curation: A.A. and A.A.A.; Writing Original Draft Preparation: A.A.A., M.M.A., and O.A.; Writing Review and Editing: O.A., M.M.A., and A.A.; Visualization: A.A.A.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] W. A. Kassab and K. A. Darabkh, "A–Z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations," *Journal of Network and Computer Applications*, vol. 2020, no. 12, pp. 1-12, 2020.

[2] Saaidah, O. Almomani, L. Al-Qaisi, N. Alsharman, and F. Alzyoud, "A comprehensive survey on node metrics of RPL protocol for IoT," *Modern Applied Science*, vol. 13, no. 1, pp. 1-15, 2019.

[3] R. Chataut, A. Phoummalayvane, and R. Akl, "Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0," *Sensors*, vol. 23, no. 16, p. 7194, 2023.

[4] O. Almomani, A. Alsaaidah, A. A. A. Shareha, A. Alzaqebah, and M. Almomani, "Performance Evaluation of Machine Learning Classifiers for Predicting Denial-of-Service Attack in Internet of Things," *International Journal of Advanced Computer Science Applications*, vol. 15, no. 1, pp. 1-10, 2024.

[5] M. Kolhar, F. Al-Turjman, A. Alameen, and M. M. Abualhaj, "A three layered decentralized IoT biometric architecture for city lockdown during COVID-19 outbreak," *IEEE Access*, vol. 8, no. 1, pp. 163608-163617, 2020.

[6] Q. Y. Shambour, M. M. Abu-Alhaj, and M. M. Al-Tahrawi, "A hybrid collaborative filtering recommendation algorithm for requirements elicitation," *International Journal of Computer Applications in Technology*, vol. 63, no. 1-2, pp. 135-146, 2020.

[7] H. F. Atlam and G. B. Wills, "IoT security, privacy, safety and ethics smart cities," in *Digital Twin Technologies*, vol. 2020, no. 1, pp. 123-149, 2020.

[8] S. Smadi, M. Alauthman, O. Almomani, A. Saaidah, and F. Alzobi, "Application layer denial of services attack detection based on stacknet," *International Journal*, vol. 2020, no. 12, pp. 2278-3091, 2020.

[9] D. Javeed, U. MohammedBadamasi, C. O. Ndubuisi, F. Soomro, and M. Asif, "Man in the middle attacks: Analysis, motivation and prevention," *International Journal of Computer Networks Communications Security*, vol. 8, no. 7, pp. 52-58, 2020.

[10] S. Zaman, "Security threats and artificial intelligence based countermeasures for internet of things networks: a comprehensive survey," *IEEE Access*, vol. 9, no. 1, pp. 94668-94690, 2021.

[11] S. Banyal and D. K. Sharma, "Security vulnerabilities, challenges, and schemes in IoT-enabled technologies," in *Blockchain Technology for Data Privacy Management*, CRC Press, vol. 1, 2021, no. 1, pp. 81-108, 2021.

[12] N. M. Karie, N. M. Sahri, and P. Haskell-Dowland, "IoT threat detection advances, challenges and future directions," in *2020 Workshop on Emerging Technologies for Security in IoT (ETSecIoT)*, Sydney, Australia, vol. 2020, no. 1, pp. 22-29, 2020.

[13] M. A. Almaiah, "Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels," *Electronics*, vol. 11, no. 21, p. 3571, 2022.

[14] O. Almomani, M. A. Almaiah, A. Alsaaidah, S. Smadi, A. H. Mohammad, and A. Althunibat, "Machine learning classifiers for network intrusion detection system: comparative study," in *2021 International Conference on Information Technology (ICIT)*, Amman, Jordan, 2021, pp. 440-445.

[15] O. Almomani, M. A. Almaiah, M. Madi, A. Alsaaidah, M. A. Almomani, and S. Smadi, "Reconnaissance attack detection via boosting machine learning classifiers," in *AIP Conference Proceedings*, Amman, Jordan, 2023, vol. 2979, no. 1, pp. 60002_1-60002_7

[16] Almomani, "Ensemble-based approach for efficient intrusion detection in network traffic," *Intelligent* Automation *Soft Computing*, vol. 37, no. 2, pp. 1-15, 2023.

 ISSN 2723-6471

[17] Heidari and M. A. Jabraeil Jamali, "Internet of Things intrusion detection systems: a comprehensive review and future directions," *Cluster Computing*, vol. 26, no. 6, pp. 3753-3780, 2023.

[18] P. K. Keserwani, M. C. Govil, E. S. Pilli, and Applications, "An effective NIDS framework based on a comprehensive survey of feature optimization and classification techniques," *Neural Computing Applications*, vol. 35, no. 7, pp. 4993-5013, 2023.

[19] S. Venkatesan, "Design an intrusion detection system based on feature selection using ML algorithms," *Mathematical Statistician Engineering Applications*, vol. 72, no. 1, pp. 702-710, 2023.

[20] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," *Symmetry*, vol. 12, no. 6, p. 1046, 2020.

[21] O. Almomani, "A Hybrid Model Using Bio-Inspired Metaheuristic Algorithms for Network Intrusion Detection System," *Computers, Materials, Continua*, vol. 68, no. 1, pp. 1-15, 2021.

[22] M. Alsharaiah, "A new phishing-website detection framework using ensemble classification and clustering," *International Journal of Data and Network Science*, vol. 7, no. 2, pp. 857-864, 2023.

[23] S. T. Revathi and N. Ramaraj, "A brief study about nature inspired optimisation algorithms," *International Journal of Advanced Intelligence Paradigms*, vol. 28, no. 1-2, pp. 1-15, 2024.

[24] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78-84, 2010.

[25] H. Xu, S. Yu, J. Chen, and X. Zuo, "An improved firefly algorithm for feature selection in classification," *Wireless Personal Communications*, vol. 102, no. 1, pp. 2823-2834, 2018.

[26] K. Chaudhary, "A Modified Firefly Algorithm for Solving Optimization Problems," *International Journal of Computational Intelligence and Applications*, vol. 2024, no. 1, pp. 1-15, 2024.

[27] M. Alsoul, H. Zaher, N. Ragaa, and E. M. Oun, "A New Efficient Hybrid Approach for Machine Learning-Based Firefly Optimization," *Iraqi Journal of Science*, vol. 2023, no. 1, pp. 4600-4612, 2023.

[28] Song, H. Qiao, Y. Huang, S. Ai, W. Xu, and H. Li, "Dynamic economic dispatch based on improved firefly optimization algorithm," in *2023 6th International Electrical and Energy Conference (CIEEC)*, Hefei, China, vol 6, no. May, pp. 1-10, 2023.

[29] Selvakumar and K. Muneeswaran, "Firefly algorithm-based feature selection for network intrusion detection," *Computers & Security*, vol. 81, pp. 148-155, 2019.

[30] S. Rajabi, S. Jamali, and J. Javidan, "An intrusion detection system in computer networks using the firefly algorithm and the fast learning network," *International Journal of Web Research*, vol. 3, no. 1, pp. 50-56, 2020.

[31] P. Ghosh, D. Sarkar, J. Sharma, and S. Phadikar, "An intrusion detection system using modified-firefly algorithm in cloud environment," *International Journal of Digital Crime Forensics*, vol. 13, no. 2, pp. 77-93, 2021.

[32] H. Mohammad, T. Alwada'n, O. Almomani, S. Smadi, and N. ElOmari, "Bio-inspired Hybrid Feature Selection Model for Intrusion Detection," *Computers, Materials, Continua*, vol. 73, no. 1, pp. 133-150, 2022.

[33] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, and H. K. Kim, "IoT network intrusion dataset," *IEEE Dataport*, Sep. 27, 2019.

[34] K. Albulayhi, Q. Abu Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method," *Applied Sciences*, vol. 12, no. 10, p. 5015, 2022.

[35] Ullah and Q. H. Mahmoud, "A scheme for generating a dataset for anomalous activity detection in IoT networks," in *Canadian Conference on Artificial Intelligence*, Ottawa, Canada (virtual), vol. 2020, no. 1, pp. 508-520, 2020.