SABADT: Hybrid Intrusion Detection Approach for Cyber Attacks Identification in WLAN

Merve Ozkan-Okay\textsuperscript{1}, Ömer Aslan\textsuperscript{2}, Recep Eryigit\textsuperscript{1}, and Refik Samet\textsuperscript{1}

\textsuperscript{1}Department of Computer Engineering, Ankara University, TURKEY
\textsuperscript{2}Department of Information Technologies, Bandırma 17 Eylül University, TURKEY

Corresponding author: Merve Ozkan-Okay (e-mail: merveozkan@ankara.edu.tr).

ABSTRACT With the advancement of technology, the use of wireless media and devices are increasing every day. In particular, the use of wireless local area networks (WLAN) has increased rapidly in recent years and is expected to increase further. The current state of wireless local area network technologies makes the network vulnerable to attacks ranging from passive listening to active intervention. Intrusion detection systems (IDSs) are being developed against these kinds of attacks. The IDSs play an important role in WLAN security by detecting and preventing malicious activities. However, most techniques used in IDSs cannot cope with dynamic and complex attacks. The aim of this study is to reduce the deficiencies in present IDSs for WLANs and build a more effective system which can detect unknown and complex attack variants dynamically. In this context, a methodology has been proposed. The proposed methodology basically has two contributions. The first contribution is the Feature Selection Approach (FSAP) to increase the speed of attack detection by reducing the number of used features. The second contribution is the hybrid attack detection technique, SABADT (Signature and Anomaly Based Attack Detection Technique), which detects attacks fast and with high accuracy. The proposed methodology is implemented on the KDD’99 and UNSW-NB15 datasets. The obtained results are compared with existing machine learning techniques. The detection model is created by using KDD’99, UNSW-NB15 training dataset and tested on the KDD’99 and UNSW-NB15 training dataset. The obtained 99.65\% and 99.17\% accuracy rates are quite high when compared to leading methods in the literature. In addition, common attack tools were used to obtain a mix of normal activities and current attack behaviors in order to test on novel attacks within the scope of the study. These different types of attacks were captured with the Wireshark tool. Some of the captured attacks were used only in the testing phase. Here, the attacks were detected with an accuracy rate of 99.69\%.

INDEX TERMS Wireless LAN, Intrusion detection system, hybrid model, signature based technique, anomaly based technique, machine learning, data analysis

I. INTRODUCTION

With the advancement of technology, cables have left most of our lives and have been replaced by wireless devices and technologies [1]. Especially, the use of wireless local area networks has increased rapidly in recent years and is expected to be increased further [2,3]. The popularity and rapid development of wireless technology has made our daily life easier, but it has brought more security problems [4,5]. For these reasons, interest in network intrusion detection systems (NIDSs) has increased among researchers [6,7]. Intrusion Detection Systems are widely used in detection of both unknown and known attacks carried out by both external and internal attackers in wireless networks. In addition, security standards for WLAN are still ambiguous for protection [8].

Due to the intensive interest in WLANs, research in the field of IDS has increased [9,10]. However, in proportion to the speed of these researches, the density and types of attacks increase as well. To illustrate, attacks may become undetectable with small changes on a known attack type [11]. Therefore, existing IDSs cannot cope with fast evolving attack surface. The goal of this study is to contribute to the development of IDS that increase the accuracy of new kinds of attacks in WLAN. In this context, a methodology consisting of four stages is proposed:

1. Datasets are selected and analyzed;
2. Features are selected and grouped;
3. Signature and anomaly based attacks are detected;
4. The training and classification are performed to detect the attacks.

While applying the proposed methodology, well known datasets in literature are analyzed such as KDD’99, AWID, ADFA-LD and UNSW-NB15[12-14]. By analyzing these datasets, the features that are commonly used when detecting attacks in networks, the characteristics of the datasets themselves, the types of attacks, etc. are examined. These datasets contain some common features such as network usage time, connection protocol, source and...
Researchers are studying various methods and datasets to detect WLAN attacks. When these studies are examined, it is concluded that there are important results that have been obtained up to the present. The summary of the related studies can be seen in Table I.

Singhrova [15] described architecture of Host-based Intrusion Detection System for DoS attack in distributed WLAN. The presented system is an intelligent system that detects the intrusion periodically and dynamically. In this context, each host monitored its surrounding nodes, and maintained a database about the transmission of its own as well as its neighbor nodes transmission. According to the authors, the proposed system is confident and efficient, as it is applied on distributed nodes and all destination IP addresses. This study can be enhanced by checking the same approach for other attack types in WLAN. Thanthri et al. [16] studied feature reduction techniques by using AWID dataset. Information Gain (IG) and Chi-Squared statistics (CH) were applied to reduce the feature number and evaluate the dataset performance. The obtained results showed that feature reduction increases the performance in terms of complexity, time and accuracy. The accuracy of classification increased 2.4% with feature reduction from 110 to 41. However, this study can be improved by implementing proposed approach on other well-known datasets.

Manzoor and Kumar [17] explained a feature reduction method which applies feature ranking based on correlation and information gain. The proposed method was based on ANN (Artificial Neural Network) and applied on KDD’99 dataset. Comparison of proposed method with and without feature reduction is performed. According to the obtained results, the performance of the feature reduction system is better than the system without feature reduction. In addition, this study can be improved further by implementing proposed approach on other datasets.

Usha and Kavita [18] presented a normalized gain based IDS for 802.11 intrusions named as NMI. The proposed NMI consists of two components. The first component is optimal feature selection and the second component is categorizing and detecting intrusions by using SVM classifier. The proposed method was applied on AWID dataset. The experimental results demonstrated that computation complexity and false positive rates were reduced by decreasing the number of features, and an accuracy rate of 99.2% was obtained. However, the proposed study can be extended by using individual attack scenarios.

Primartha and Tama [19] explained a random forest classifier by setting parameters to enhance the performance of anomaly detection in the IoT network. They compared IDSs performance with the proposed algorithm by using false alarm rate and accuracy metrics. They experimented on the GPRS, UNSW-NB15 and NSL-KDD dataset. According to the authors, they obtained better results than most recent studies on the NSL-KDD dataset in terms of false alarm rate and accuracy. Also, obtained a better result than the MLP algorithm on the GPRS dataset in terms of accuracy, but could not perform better on the UNSW-NB15 dataset. Comparison of NIDSs is explained by Magán-Carrión et al. [22] The study consists of two stages. First stage includes a
### Table I. Summary of related studies on IDS

| Paper | Proposed Method | Goal/Success | Year |
|-------|-----------------|--------------|------|
| Singhrova [15] | Host based intrusion detection system for DoS attack. Feature reduction techniques such as Information Gain (IG) and Chi-Squared statistics (CH). | It is confident and efficient, as it is applied to distributed nodes. Feature reduction can give better results according to accuracy, time and complexity. | 2011 |
| Thanthrige et al. [16] | Feature reduction methods such as Information Gain (IG) and Chi-Squared statistics (CH). | Accuracy rates of normal and DoS, u2r, r2l, probe attacks were obtained: 94.80%, 99.93%, 96.51%, 99.54%, and 98.79%, respectively. | 2016 |
| Manzoor and Kumar [17] | Feature reduction method based on information gain and Artificial Neural Network. | The computation complexity and false positives reduced and accuracy rate of 99.2% was obtained. | 2017 |
| Usha and Kavita [18] | Gain based intrusion detection system for 802.11 intrusions named as NMI. | Obtained better results than most recent studies on NSL-KDD dataset according to false alarm rate and accuracy. | 2017 |
| Primartha and Tama [19] | Random forest classifier with parameter setting. | | |
| Abdulhammed et al. [20] | Proposed feature selection and multiclass classification. | Random Forest algorithm and 32 features achieves a maximum accuracy of 99.64%. | 2018 |
| Zwane et al. [21] | Analyzed performance of machine learning classifiers for IDSs. | Ensemble-based learning methods are better than single learning methods. However, it can be slower in terms of building time and model testing time. | 2018 |
| Magán-Carrión et al. [22] | Including comprehensive research and proposed a methodology named as FaaC for addressing NIDSs. | Investigated papers did not accomplish mandatory steps for a reliable evaluation. The proposed methodology can help future research to fairly assess NIDSs. | 2020 |
| Lopez-Martin et al. [23] | Proposed an application of deep reinforcement learning (DRL) algorithms for intrusion detection. | Double Deep Q-Network algorithm gave the best result. The proposed classifier with DRL is faster than other models. | 2020 |

A comprehensive research of recent network based intrusion detection systems based on machine learning approaches. Second stage includes a methodology named as FaaC (Feature as a Counter) for addressing network attack detection problems. The proposed methodology is tested on UGR’16 dataset. According to the authors, almost all of the investigated papers did not accomplish mandatory steps for a reliable comparison and evaluation of NIDSs. However, this study can be further improved not only by using standard machine learning techniques, but also proposing their own method.

According to Abdulhammed et al. [20] feature selection is a key factor for an improved wireless intrusion detection system based on machine learning classifiers. This study discusses multiclass classification using four effective feature sets of 5, 7, 10, and 32 features, respectively. The obtained results used the AWID dataset to evaluate the efficiency of seven well-known machine learning classifiers based on the selected set of features. The proposed system used a Random Forest algorithm with 32 features and achieved a maximum accuracy of 99.64%. Zwane et al. [21] analyzed performance of machine learning classifiers for IDSs. In this paper, different seven classifiers based on machine learning are analyzed. The WEKA was used to evaluate and implement the classifiers. According to the obtained results, ensemble-based learning methods are better than the single learning methods. However, ensemble classifiers can be slower in terms of building and testing time.

Lopez-Martin et al. [23] suggested an application of different deep reinforcement learning (DRL) algorithms to intrusion detection by using NSL-KDD and AWID datasets. They made a conceptual change to the classical DRL paradigm. In addition, they presented the results of applying their technique to four DRL models: Actor-Critic, Double Deep Q-Network, Deep Q-Network and Policy Gradient. According to the obtained results, the Double Deep Q-Network algorithm produced the best result. Additionally, the proposed classifier by using DRL is faster than other models.

Duan et al. [24] suggested an intelligent intrusion detection framework which uses deep belief networks (DBN) and principal component analysis (PCA). They used two different methods to preprocess the data, namely: PCA with BP (back propagation), and DBN. In PCA with BP stage, they used PCA to perform feature learning on the collected network data, and combined with BP as a classifier to complete data preprocessing steps. In the DBN stage, they extracted original input layer by layer and combined it with multi-layer RBM (restricted Boltzmann machines). According to the paper, the test results showed that DBN improved the performance for feature learning. A wireless intrusion detection system (WIDS), which works on deep learning with a wrapper, is proposed by Kasongo and Sun [25]. The proposed system used a feed-forward deep neural network (FFDNN) which uses wrapper-based feature extraction (WFE). The WFE uses additional trees to generate an optimal feature vector. To evaluate the model performance, they used UNSW-NB15 and the AWID datasets. They also compared their results with RF, SVM, NB, DT, and KNN classifiers. According to the test results, the proposed system outperformed the existing approaches in terms of accuracy.

When existing studies are examined, it is seen that well-known datasets in the literature such as KDD’99, AWID etc. are used. The various techniques such as preprocessing feature selection and reduction, and machine learning methods have been applied on datasets to detect attacks and obtain high accuracy. When the techniques in these studies are evaluated, it is seen that preprocessing and feature reduction steps increase performance before applying machine learning methods [10,26-28]. Additionally, some machine learning methods may outperform others depending on the number of used features, the distribution and size of the data generated [5]. As a result, when studies are
examined, it seems that WLAN attacks still cannot be detected dynamically with high accuracy and there is no classification for attack types. In addition, studies only work on feature reduction or attack detection. There is no study that presents an original approach for both. In addition, when testing methodologies, studies generally used a single data set. In this context, our proposed study conducts experiments on more data sets. At the same time, a small data set was created using attack tools, and a study was conducted on it. With these aforementioned aspects, our study provides more important contributions than the studies in the literature.

### III. THE PROPOSED METHODOLOGY

It is important to detect attack traffic with high accuracy to protect WLANs. In addition to detecting with high accuracy, fast detection is also important. In this context, a methodology is proposed to detect attacks quickly and accurately. Although the time and memory usage of the signature based model is low, the anomaly based model provides high accuracy thanks to its ability to detect previously unknown attacks.

The block diagram of the proposed method is given in Figure 1. The proposed method consists of 4 blocks:

1. Dataset is selected and analyzed
2. Features are selected and grouped
3. Signature and anomaly based attacks are detected
4. The training and classification are performed to detect the attacks

In Block 1, the features are grouped according to the traffic results (normal/attack) to determine the effect of each feature. Existing datasets and analysis methods were used in scope of Block 1. One of the main contributions of study is the FSAP approach for feature selection which was proposed in scope of Block 2. To select the important features, the calculated threshold value and proposed algorithm are used. Another contribution of study is SABADT which is a combination of signature and anomaly based techniques for attack detection which was proposed in scope of Block 3.

Firstly, signatures are extracted with features that can be reached to the result directly by using signature-based technique. Secondly, behaviors are determined and rules are extracted by using anomaly-based technique. In Block 4, the training and classification are carried out with both the proposed methodology and known techniques in the literature.

#### A. ANALYZING DATASET

At this stage, well-known datasets including KDD’99, UNSW-NB15, ADFA-LD, AWID etc. were used. These datasets were analyzed according to their features and results. As a result of the analysis, common features of datasets were determined. These common features are used in the signature-based model stage of the proposed attack detection technique. The content features of the datasets are used in the anomaly-based model stage of the proposed attack detection technique. The features of 3 different dataset are given in Table II. There are some features common to all of them. Some of these features are marked and these features are used as basic features that can be in every data set.

#### B. PROPOSED FEATURE SELECTION APPROACH (FSAP)

Feature selection process is one of the most important steps in building a machine learning model [29]. Each feature also has an impact on the result of traffic, but some of these features have less impact [30, 31]. Proposed FSAP approach for feature selection consists of two steps. In the first step, threshold value is calculated by using Eq. (1).

$$ t = \frac{\sum_{i=1}^{f} B_{i} \cdot \text{std}(g)}{g} $$  \hspace{1cm} (1)$$

Where, $t$ is the threshold value for feature selection, $f$ is the total number of features in the dataset and $g$ is the number of groups formed for the “feature & traffic result” grouping, $\text{std}(g)$ is the standard deviation of all groups.
As a result of Eq.1, the features below the threshold value were eliminated. In the second step, the groups formed by the remaining features (> t) were analyzed. The number of data in the group can be above the threshold value. However, the feature may consist of a wide range of values or each value can correspond to both traffic results (normal or attack). For example, suppose a feature consists of 100 (1, 2, ..., 100) different values. If each value meets both the attack and the normal situation, a group may consist of 200 different situations. The number of groups may remain above the threshold value, but when the content is analyzed, it is not an important feature in detecting the attack status. Formation of a group is given in Table III.

| Table II. Features of different datasets |
|-----------------------------------------|
| **KDD’99 features** | **UNSW-N15 features** | **ADFA-LD features** |
| duration | serr_rate | dur | smean | srcip |
| protocol_type | srv_rerror_rate | proto | dmean | sport |
| service | same_srv_rate | service | trans_depth | dstip |
| flag | diff_srv_rate | state | response_body_len | dport |
| src_bytes | srv_diff_host_rate | spkts | ct_srv_srch | dstip |
| dst_bytes | dst_host_count | dpkts | ct_state_ttl | state |
| land | dst_host_srv_count | sbytes | ct_dst_ltm | dur |
| wrong_fragment | dst_host_same_srv_rate | dbytes | ct_src_dport_ltm | sbytes |
| urgent | dst_host_diff_srv_rate | rate | ct_dst_sport_ltm | dbytes |
| hot | dst_host_same_src_port_rate | sttl | ct_dst_sport_ltm | sttl |
| num_failed_logins | dst_host_srv_diff_host_rate | dttl | is_fip_login | dttl |
| logged_in | dst_host_serror_rate | sload | ct_fip_cmd | sload |
| num_compromised | dst_host_srv_serror_rate | dload | ct_flw_http_mthd | dloss |
| root_shell | dst_host_rerror_rate | sloss | ct_src_ltm | service |
| su_attempted | dst_host_srvr_rerror_rate | dloss | ct_srv_dst | sload |
| num_root | dst_host_srvr_rerror_rate | sttl | is_sm_ips_ports | dload |
| num_file_creations | dst_host_serror_rate | dtime | is_fip_login | dtime |
| num_shells | dst_host_serror_rate | dtime | ct_fip_cmd | dtime |
| num_access_files | dst_host_serror_rate | dtime | ct_flw_http_mthd | dtime |
| num_outbound_cmds | dst_host_serror_rate | dtime | resp_body_len | dtime |
| is_host_login | dst_host_serror_rate | dtime | ct_src_ltm | dtime |
| is_guest_login | dst_host_serror_rate | dtime | ct_dst_ltm | dtime |
| count | dst_host_serror_rate | dtime | tcpp | dtime |
| srv_count | tcpport | dtime | dmeanz | dtime |
| serror_rate | dst_host_serror_rate | dtime | srcip | dstip |
| srv_serror_rate | dst_host_serror_rate | dtime | proto | dstip |

As a result of Eq.1, the features below the threshold value were eliminated. In the second step, the groups formed by the remaining features (> t) were analyzed. The number of data in the group can be above the threshold value. However, the feature may consist of a wide range of values or each value can correspond to both traffic results (normal or attack). For example, suppose a feature consists of 100 (1, 2, ..., 100) different values. If each value meets both the attack and the normal situation, a group may consist of 200 different situations. The number of groups may remain above the threshold value, but when the content is analyzed, it is not an important feature in detecting the attack status. Formation of a group is given in Table III.

| Table III. Feature groups |
|---------------------------|
| f - f1, f2, f3, ..., fL, list of features. The list consists of L different features. |
| f1 - [x1, x2, x3, ..., xK], each feature consists of K different values. |
| β - [0, 1] - 0 indicates normal and 1 indicates attack. |
| τ - [τ1, τ2, τ3, ..., τN] in case of (τK, β) is the number of corresponding attacks or normal states. The sum of these values gives the total number of data. |
| g - [g(f1(x1, β)), g f1(x2, β), g f1(x3, β), ..., g f1(xK-1, β), g f1(xK, β)], it is the group formed by separating the values in a feature according to the attack/normal state. |

In the second step of feature selection, following parameters are determined:
- the number of different data in the group
- the distribution of the number of data in the group on the result
- the min/max/average values of normal
- attack states according to the features

Features that provide the result of these parameters are added to the selected properties list. The used algorithm when selecting the features is given in Algorithm 1.
ALGORITHM 1: FSAP Algorithm

**Input:** Feature Groups List

**Output:** Selected Features List

**Definitions:**

\( r \): the number of results corresponding to the data in the group

\( \text{T}_N \): total number of normal results corresponding to the data in the group

\( \text{T}_A \): total number of attack results corresponding to the data in the group

\( \beta \): state corresponding to the data in the group

\( \mu \): the same number of data in group

\( \text{m}_N \): normal max value of the feature

\( \text{m}_A \): max value of the feature in attack state

\( \text{m}_{\beta} \): min value of the feature in attack state

1. \( g_1 \leftarrow \text{group1}, g_2 \leftarrow \text{group2}, \ldots, g_N \leftarrow \text{groupN} \) \( s \leftarrow u(g(j)) \)
2. for \( i \leftarrow 1 \) to \( n \) do
3.     for \( j \leftarrow 1 \) to \( s-1 \) do
4.         if \( \beta(g(i)) = \text{normal' then} \)
5.             \( \text{T}_N = \text{T}_N + r(g(i)) \)
6.         else \( \mu = \text{g}(i+1) \)
7.             \( \text{m}_N = \text{g}(i+1) \)
8.         end if
9.     end for
10.    if \( \mu > \text{m}_A \) then
11.        \( \text{T}_A = \text{T}_A + r(g(i)) \)
12.    else \( \mu = \text{g}(i+1) \)
13.        \( \text{m}_A = \text{g}(i+1) \)
14.    end if
15.    if \( \mu = \text{g}(i+1) \) then
16.        \( \mu = r(g(i))/ r(g(i+1)) \)
17.        \( \text{H}=\frac{\mu}{\mu} \)
18.        do
19.            if \( g(i) > \text{g}(i+1) \)
20.                \( \mu = r(g(i))/ r(g(i+1)) \)
21.            end if
22.        end for
23.        add(f)
24.    else eliminate feature
25.    end if
26.    if \( \text{m}_N < \text{m}_A \) or \( \text{m}_A < \text{m}_N \) then
27.        eliminate feature
28.    else if \( \text{m}_N = \text{m}_A \) or \( \text{m}_A = \text{m}_N \) then
29.        eliminate feature
30.    else add(f)
31.    end if
32. end for
33. end for

As a result of applying threshold value and feature selection algorithm, important features were selected. Selecting the appropriate features is very important for creating a successful model in machine learning. In this context, with the FSAP approach, both the selection of the correct features and the feature reduction take place.

C. PROPOSED HYBRID SIGNATURE AND ANOMALY BASED ATTACK DETECTION TECHNIQUE (SABADT)

The SABADT model is proposed by combining the signature and anomaly based techniques. In the first step, signature based technique has been used. Because this technique quickly detects known attacks. In the second step, anomaly based technique has been used as it can also detect unknown attacks. With this hybrid model, it is aimed to develop an attack detection system that is fast and has high accuracy.

1) SIGNATURE BASED MODEL

At this step of the model, firstly, the behaviors (signatures) of the attack types known in the literature are considered. Their rules are recorded. Some of these known rules are as follows:

1. Flood attacks often cause increase in management frameworks per unit time. While it is easy to distinguish a flood attack from normal traffic, it is not always easy to distinguish it from other types of attacks [32-34].

2. Injection attacks often cause corruption of valid and encrypted data frames. In these attacks, the attacker wants to transfer a large number of small data frames over a considerable period of time [35].

3. During fragmentation attack, the attacker injects a short and fragmented data frame. If successful, this process usually takes no more than a second, but if not successful, the same process is repeated [36].

4. Impersonation attacks provide an additional AP (access point) in the neighborhood broadcasting a pre-existing valid network. The common signature of all impersonation attacks is that the victim network nearly doubles the number of Beacon frames [37].

Afterwards, datasets commonly used in the literature were examined [9,38,39]. Each dataset created has its own content features as well as basic features that are commonly used when detecting intrusions. Rules are extracted based on these basic features. Firstly, each of the selected based features was analyzed separately. Parameters were determined from the result of this analysis (average/max/min normal time, number of transferred packets, used protocols, etc.). If there are cases in which the result can be reached directly, the rules were extracted and added to the signature list. For instance, the maximum time in the normal state of a network was determined and a rule was extracted by comparing time in the attack state. Secondly, all of the selected based features were analyzed together. This is because situations other than the determined parameters are prevented from being ignored. For example, in case of an attack, the usage time of the network can be in the normal range. When the time feature is examined by adding the amount of transferred data feature, the amount of transferred data may be much higher than normal status. This shows that there is an attack in normal traffic time. The aim of the first step of the model is to determine the results that can be reached directly and to increase the speed of detecting the attack. In the second step, it is aimed to obtain a high accuracy rate by examining all situations that can be overlooked. The algorithm of the signature-based model is given in Algorithm 2. (Due to the large number of parameters and rules determined, the calculation details are not given for each of them, they are written as basic functions).
To summarize in general, both the signatures of known attacks in the literature and the signatures of unknown attacks determined on the common basic features in widely used datasets were recorded. Initially, each of the selected basic features was analyzed separately and the parameters were determined as a result of the analysis. If there are situations where the result can be reached directly, the rules were extracted and added to the signature list. Then, all of the selected basic features were analyzed together. With this model, it is aimed to increase the accuracy rate by detailed analysis of the features and the speed of detection of the attack with rules reached to result directly.

2. ANOMALY BASED MODEL

In the second step of the model, traffic behavior analysis was performed by using the content features of the dataset. The time-dependent traffic features, additional features according to the dataset, and the rules were extracted. Thanks to the anomaly-based model, unknown attacks can be detected. Although the time and memory usage is high compared to the signature-based model, it provides high accuracy thanks to its ability to detect previously unknown attacks [40,41]. While creating an anomaly-based model, each feature group was first analyzed among themselves. After that, all the features were analyzed together and the rules were extracted. The algorithm of the anomaly-based model is given in Algorithm 3 (Due to the large number of parameters and rules determined, the calculation details are not given for each of them, they are written as basic functions).

**Algorithm 2: Signature Based Algorithm in SABADT**

**Input:** \( f \): Selected based features  
**Output:** \( \beta \): Determined signatures (rules)  
**Definitions:**  
\( \beta \): corresponding status in the network  
\( NDP \): Normal state parameters  
\( ADP \): Attack status parameters  
\( r_1 \): extracted rules  

\[
\begin{align*}
1. & \quad f_1 \leftarrow \text{feature1}, \ldots, f_n \leftarrow \text{featureN}, \quad n \leftarrow u(f(i)) \\
2. & \quad \text{for } i = 1 \text{ to } n \text{ do} \\
3. & \quad \quad \text{if } \beta(f(i)) = 'normal' \text{ then} \\
4. & \quad \quad \quad \text{write.NDP()} \\
5. & \quad \quad \text{else} \\
6. & \quad \quad \quad \text{write.ADP()} \\
7. & \quad \quad \text{end if} \\
8. & \quad \text{end for} \\
9. & \quad \text{compare(NDP, ADP)} \\
10. & \quad \text{determine.r()} \\
11. & \quad \text{write.s(r)} \\
12. & \quad \text{analyze.f()} \\
13. & \quad \text{determine.r()} \\
14. & \quad \text{write.s(r)}
\end{align*}
\]

**Algorithm 3: Anomaly Based Model Algorithm in SABADT**

**Input:** \( f \): Selected content features  
**Output:** \( \beta \): Determined signatures (rules)  
**Definitions:**  
\( \beta \): corresponding status in the network  
\( ND P \): Normal state parameters  
\( AD P \): Attack status parameters  
\( r_2 \): extracted rules  

\[
\begin{align*}
1. & \quad f_1 \leftarrow \text{feature1}, f_2 \leftarrow \text{feature2}, f_n \leftarrow \text{featureN}, \quad n \leftarrow u(f(i)) \\
2. & \quad \text{for } i = 1 \text{ to } n \text{ do} \\
3. & \quad \quad \text{if } \beta(f(i)) = 'normal' \text{ then} \\
4. & \quad \quad \quad \text{write.NDP()} \\
5. & \quad \quad \text{else} \\
6. & \quad \quad \quad \text{write.ADP()} \\
7. & \quad \quad \text{end if} \\
8. & \quad \text{end for} \\
9. & \quad \text{analyse.f()} \\
10. & \quad \text{determine.r()} \\
11. & \quad \text{write.s(r)} \\
12. & \quad \text{compare(NDP, ADP)} \\
13. & \quad \text{determine.r()} \\
14. & \quad \text{write.s(r)}
\end{align*}
\]

In the anomaly-based model, the number of rules is large and the details are quite complex. During the implementation of the model, the use of time is high, but the accuracy rate increases when detecting the attacks. In addition, thanks to the reduced features, improvement is achieved in terms of time reduced features.

**D. TRAINING AND CLASSIFICATION**

While performing the training and classification, the proposed methodology and machine learning techniques in the literature were applied on KDD’99 dataset. While applying the training phase, the KDD’99 training dataset (494020) was used and applying testing phase, the KDD’99 testing dataset (311028) was used. As well as machine...
learning algorithms have been used in many different areas for many years, they have been used in the analysis and detection of attacks in wireless local area networks recently. Therefore, applicable algorithms are used in this study. Although it cannot be said that one algorithm is better than another in general, each algorithm has its own advantages and disadvantages. An algorithm can perform better than other algorithms depending on the distribution of the data, the number of features, and the dependencies between features [15,19,42,43].

E. EVALUATION OF MODEL PERFORMANCE

To evaluate the model performance as well as comparing the machine learning algorithms performance, metrics such as recall, false positive, false negative, precision, f-measure and accuracy are used. These values are calculated using the confusion matrix (Table IV).

| Table IV. Confusion Matrix |
|---------------------------|
| Actual  |  Predicted  |
|         | Positive | Negative |
| Positive | TP       | FN        |
| Negative | FP       | TN        |

Detection rate = TP/(TP+FN) (1)
False positive = FP/(FP+TN) (2)
Precision = TP/(TP+FP) (3)
f-Measure = (2*precision*recall)/(precision+recall) (4)
Accuracy = (TP+TN)/(TP+TN+FP+FN) (5)

IV. IMPLEMENTATION OF PROPOSED METHODOLOGY

Implementation of the proposed methodology was carried out on the Python platform on KDD’99 dataset. The implementation on KDD’99 dataset has been explained since it contains a high amount of data and is the most used dataset in this field. The implementation of this methodology is explained in detail in the following sections.

A. KDD’99 DATASET

The KDD’99 dataset was created by passing the DARPA data set through some preprocessing in 1999 and has been used and cited in many studies [38]. The KDD’99 dataset is primarily used in IDS and machine learning research. In the KDD’99 data set, a feature map consisting of 9 basic and 32 derived, in total 41 features, was created. These 41 features are defined in four main categories: Basic features, Content features, Host-based traffic features and Time-based traffic features (Table V).

The basic features of this dataset are the basic features taken only from TCP connections. These features are easier to achieve than other categories as there is no need to preprocess the network traffic data. Traffic features are the link content features that appear with domain information. Time-dependent traffic features are the names given to features extracted using “same server” and “same service” features. “Same server” features are obtained by examining the connections made to the same server in the last two seconds. Similarly, “same service” features are obtained by examining the connections made to the same service in the last two seconds.
KDD’99 dataset includes 38 different statuses in total. While 23 of these are in the training dataset, the test dataset includes 15 more attack types that are not included in the training dataset. In this study, it is aimed to detect known and unknown 15 attack types with high accuracy (Table VI).

| Attack Type in KDD’99 Training Dataset | Unknown Attack Type in KDD’99 Testing Dataset |
|----------------------------------------|-----------------------------------------------|
| back                                   | apache2                                       |
| buffer_overflow                        | httpbomb                                     |
| ftp_write                              | mscan                                        |
| guesspasswd                            | named                                        |
| imap                                   | processable                                   |
| ipsweep                                | saint                                        |
| land                                   | sendmail                                      |
| loadmodule                            | snmpgetattack                                 |
| multihop                               | snmpguess                                     |
| neptune                                | sqlattack                                     |
| nmap                                   | udpstorm                                      |
| normal                                 | worm                                          |
| perl                                   | xclock                                        |
| phf                                    | xsnoop                                        |
| pod                                    |                                               |
| portsweep                              |                                               |
| rootkit                                |                                               |
| Satan                                  |                                               |
| snarf                                  |                                               |
| spy                                    |                                               |
| teardrop                               |                                               |
| warezclient                            |                                               |
| warezmaster                            |                                               |

While creating the proposed technique, KDD’99 training dataset was used. In order to determine traffic results, all attack types are labeled as "attack" and "normal". First of all, the important features are selected with the proposed feature selection approach. Afterwards, the model was created by extracting rules from the situations that can directly reach the result (signature-based) and traffic behavior (anomaly-based). In the testing phase of the proposed model, the whole KDD’99 dataset was used.

1. IMPLEMENTATION OF FSAP

Firstly, each feature in the dataset is grouped with the traffic condition result (normal/attack) (service-result, count-result, duration-result, etc.). The number of groups formed by each feature, that is, its effect on the result, is compared with the calculated threshold value. Features above this value are important features for the result and these features have been selected.

In Figure 2, when the "logged_in" feature is separated according to the values it contains and the attack status, it has created 4 different groups but its determination is low. When the "logged_in" feature is 0, there are 393445 attacks, 27339 normal situations, and when it is 1, there are 3298 attacks, 66939 normal situations. Looking at the numbers, the number of attacks and the number of normal traffic is not negligible in the same situation. When considered together with other situations, there is of course a determining aspect, but there are more important features that can make us reach results faster. For this reason, this feature, which is below the threshold value, has been eliminated while selecting the features.

![Figure 2. Grouping result of "logged_in" feature](image)

2. IMPLEMENTATION OF SABADT

By using the remaining features after the feature selection algorithm, one of the basic features of the KDD’99 dataset specified in Table 4, the rules that can directly achieve the result of the attack for the signature-based model have been extracted. These were recorded by labeling "attack". In this stage, the result of 35% of the dataset can be reached. In this case, it saves us time and memory. Some of these signatures are given in Table VIII.
In the second step of the proposed model, the behavior analysis has been performed and rules have been created by using the content features, host-based features and time-based features of KDD’99. During the analysis, the first content features, then the host-based features, and then the time-based features were examined, sequentially. Some of the rules are as follows: if making a connection over a private service type, no data flow is seen and the number of failed logins is higher from 0. If the number of connections made to the same host in the last 2 seconds is intense, etc., these are labeled as an attack. Some of these rules are given in Table IX.

In the last stage, the rules derived by using KDD’99 training dataset were applied to the KDD’99 testing dataset.

| Service=private, src-bytes=0, dst-bytes=0 | Attack |
|------------------------------------------|--------|
| Service=private, serror-rate >= 0, serror-rate <= 0 | Attack |
| Service=eco-i, srv-diff-host-rate = 1 | Attack |
| Service=http, src-bytes=max(“normal-src-bytes”) | Attack |
| Service=finger, count=max(“normal-count”), dst-host-srv-host=srv-attack-rate >= 0.1 | Attack |
| Service=ssh, src-bytes=0, dst-bytes=0 | Attack |
| Service=telnet, duration=0, dst-host-same-srv-rate=1, dst-host-diff-srv-rates=0, dst-host-same-srv-port-rate=1 | Attack |
| Service=telnet, count>max(“normal-count”), srv-count > max(“normal-srv-count”) | Attack |
| Other situations without attack | Normal |

Table VIII. Extracted signatures from basic features (The list is abbreviated)

Table IX. Extracted rules from additional features (The list is abbreviated)

To evaluate the success of the proposed methodology when detecting unknown attacks, KDD’99 testing dataset which consists of 311028 records was used. Test dataset includes 15 more attack types that are not included in the training dataset. In addition to the proposed methodology, different machine learning techniques in the literature were also used. These are Naive Bayes, SVM, Decision Tree, and Decision Table.

4. Obtained Results

The results of the metrics used to compare the performances of the proposed model are given in Table X.

As a result of testing the proposed method, an accuracy rate of 99.65% was achieved in detecting attacks. SVM and Decision Tree techniques have also yielded good results, but their performance in terms of time and memory usage is poor compared to the proposed methodology.

The leading methods which are used in the literature can be seen in Table XI. Considering the methods in the literature, there are not many studies that have developed algorithms for both feature selection and attack detection. Compared to the existing ones, the study of Babirye and Mwebaze [51] produced the closest accuracy to the proposed method, but the proposed method is more successful in terms of both the number of used features and the obtained accuracy rate. Although the study of Iwendi et al. [53] seems effective in terms of the number of used features, the proposed method is more successful in terms of accuracy rate.

| Paper | Feature selection | Classification technique | Number of features | Accuracy rate (%) |
|-------|-------------------|--------------------------|-------------------|------------------|
| Xiao and Liu 2009 [4] | Eliminating and removed the | SVM | 21 | 86.3 |
| Araújo et al. 2010 [44] | Redundant features Hybrid approach: k-means and gain ratio | K-means | 14 | 90.00 |
| Amiri et al. 2011 [45] | Comparing three different selection methods | Least squares support vector machine | FFS:29 | SVM:22 | LCFS:36 | Accuracy: 97.92 |
| Li et al. 2012 [46] | Gradually feature removal method | SVM | 19 | 98.62 |
| Karimi et al. 2013 [47] | Hybrid filtering feature selection | Naive Bayes | 16 | 98.28 |
| Saxena and Richiarya 2014 [48] | Standard information gain | Hybrid Approach | PSO-SVM | 18 | 99.4 |
| Olusola et al. 2015 [49] | Dependency ratio | | | 34 | - |
| Hasan et al. 2016 [50] | Permutation importance measure | Random Forest | 25 | 82.36 |
| Manzoor and Kumar 2017 [17] | Information gain and correlation based on ANN | 25 | 97.91 |
| Babirye and Mwebaze 2018 [51] | Standard information gain | Genetic algorithm | 21 | 99.64 |
| Kaur and Singh 2019 [52] | - | D-Sign based on Deep Learning | Bagging and AdaBoost classifier | SABADT | 17 | 99.65 |

Proposed Technique FSAP
B. UNSW-NB15 DATASET

The raw network packets of the UNSW-NB15 dataset were generated by the IXIA PerfectStorm tool at the Australian Cyber Security Center's (ACCS) Cyber Range Lab to yield a mix of real normal activities and synthetic contemporary attack behavior. The tcpdump tool was used to capture 100 GB of raw traffic (for example, Pcap files). It consists of 49 features in total (Table XII). There are nine different types of attacks in this dataset: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms (Table XIII).

### TABLE XII. UNSW-NB15 FEATURES

| Flow features | Based features | Content features | Time features | Additional features | Labeled features |
|---------------|----------------|-----------------|---------------|-------------------|-----------------|
| srcip         | stcp            | state           | state         | ct_state_ttl      | attack_cat      |
| sport         | dur             | dwin            | djit          | ct_flw_http_mthd | label           |
| dstip         | dbytes          | scp             | Stime         | is_flt_login      |                 |
| dsport        | dbytes          | dscp            | dtime         | ct_flt_cmd       |                 |
| proto         | stl             | xmeanz          | ampkt         | ct_srv_sec       |                 |
| dttl          | dmeanz          | dstp            | ctt_srv_dst   |                  |                 |
| dloss         | dloss           | trans_depth     | ct_sce_lim   |                  |                 |
| dloss         | dloss           | res_bdy_len     | ctt_dst_lim   |                  |                 |
| service       | ackdat          | ctt_src_port    | ctt_dst_port  |                  |                 |
| sload         | ttt_stm_op      | is_sm_ip        | ctt_dst_port  |                  |                 |
| dload         | spkts           |                 | ctt_dst_src   |                  |                 |
| dpkts         |                 |                 |               |                  |                 |

UNSW-NB15 dataset features are classified into six different groups as follows:

1. Flow features: These features have identifying features between hosts such as client-to-server or server-to-client.
2. Basic features: These features contain features that represent protocol links.
3. Content features: These features include features of TCP/IP; they also include some features of http services.
4. Time features: This group contains time features such as the arrival time between packets, packet start/end time, and round-trip time of the TCP protocol.
5. Additional created features: This group can be further divided into two groups: (1) General-purpose features where each feature has its own purpose to protect the service of the protocols. (2) Connection features are generated from a stream of 100 record connections in sequential order of the last time feature.
6. Labeled features: This category represents the label of each record.

### TABLE XIII. ATTACK TYPES IN UNSW-NB15 DATASET

| Type of attack | Definition                                                                 |
|----------------|----------------------------------------------------------------------------|
| fuzzers        | It is an attack by an attacker by feeding an application, operating system, or network security holes with large amounts of random data input to make it crash. |
| analysis       | Ports are a type of intrusion that infiltrates web applications via emails and web scripts. |
| backdoors      | It is a technique of finding plain text input while dealing with actions such as bypassing a secret authentication, unauthorized remote access to the device, and going unnoticed. |

The UNSW-NB15-training-set dataset was used while creating the model for training and classification. The attack types are labeled as "attack" and normal situations are labeled as "normal" in order to determine the result as attack or normal, which is the first stage of our problem. First of all, important features were selected with FSAP. Afterwards, our model was created by extracting the rules from the situations (signature-based) and the behavior of the traffic (anomaly-based) where we could directly reach the result. In the testing phase of the model, the UNSW-NB15_testing-set dataset was used.

1. IMPLEMENTATION OF FSAP

Firstly, each feature in the dataset is grouped with the traffic condition result (normal/attack) (service-attack_cat, proto-attack_cat, stdl-attack_cat etc.). The number of groups formed by each feature, that is, its effect on the result, is compared with the calculated threshold value. Features above this value are important features for the result and these features have been selected.

In Figure 4 "ct_flt_cmd" feature consists of 0.1 and 2 values. When the "ct_flt_cmd" feature is separated according to the values it contains and the attack status, it has created 6 different groups but its determination is low. When the "ct_flt_cmd" feature is 0, there are 45021 attacks, 36661 normal situations. Looking at the numbers, the number of attacks and the number of normal traffic is not negligible in the same situation. When considered together with other situations, there is of course a determining aspect, but there are more important features that can make us to reach results faster. For this reason, this feature, which is below the threshold value, has been eliminated while selecting the features.
FIGURE 4. Grouping result of "ct_ftp_cmd" feature

In Figure 5, when the "spkts" feature is separated according to its values and the attack status, it has formed 572 different groups. There is a heterogeneous distribution in general when looking at the types of services and attack types that are connected. For instance, while the packet number is 2, there are 2 different situations and the number of "attacks" is significantly higher than normal. After a certain value, that is, as the amount of packets increases, the attack situation becomes more pronounced. The parameters specified in the proposed feature selection method are also provided in this feature. As a result, the "spkts" feature has been added to the feature list, both by being above the threshold value and providing feature selection methods.

FIGURE 5. Grouping result of "spkts" feature

In summary, the list of selected features to be used in the created model as a result of the proposed feature selection method is given in Table XIV. 13 features were selected from 49 features with FSAP.

2. IMPLEMENTATION OF SABADT

By using the remaining features after the feature selection algorithm, one of the basic features of the UNSW-NB15 dataset specified in Table XII, the rules that can directly achieve the result of the attack for the signature-based model have been extracted. These were recorded by labeling "attack". In this stage, the result of 40% of the dataset can be reached. In this case, it saves us time and memory. Some of these signatures are given in Table XV.

| TABLE XIV. SELECTED FEATURES |
|-------------------------------|
| service | proto | duration | sttl | spkts | dkpts | sbytes | dbytes | ct_dst_src_ltm | ct_dst_sport_ltm | ct_src_dport_ltm | sloss | response_body_len |

| TABLE XV. EXTRACTED SIGNATURES FROM BASIC FEATURES (THE LIST IS ABBREVIATED) |
|-----------------------------------------------------------------------------|
| min("normal_duration")<duration<max("normal_duration") | normal |
| service="smtp", protocol="tcp", dur> max("normal_duration") | attack |
| min("normal_src_bytes")<sbytes <max("normal_src_bytes") | normal |
| service="dns", protocol="udp", dur > max("normal_duration") | attack |
| duration=0, spkts>max(normal_spkts) | attack |

In the second step of the proposed model, the behavior analysis has been made and rules have been created by using the content features, flow features, time features and additional features of UNSW-NB15. While analyzing, first the content features, then the flow features, then the time features, and then the time-based features were examined, sequentially. Some of these rules are given in Table XVI.

| TABLE XVI. EXTRACTED RULES FROM ADDITIONAL FEATURES (THE LIST IS ABBREVIATED) |
|-------------------------------------------------------------------------------|
| sttl=max(normal_dest_time) & response_body_len>0 | attack |
| sttl=max(normal_dest_time) & ct_dst_src_ltm<min(normal) & sbytes<avg(normal) | attack |
| sttl=max(normal_dest_time) & min(normal)<ct_dst_src_ltm=max(normal) & min(normal)<sbytes<avg(normal) | normal |
| ct_dst_src_ltm>max(normal) & response_body_len >max(normal) & ct SRC_dport_ltm>max(normal) | attack |
| dur>min(normal) & response_body_len=0 & tcprtt<min(normal) | attack |

In the last stage, the rules extracted using the training dataset were applied to the test dataset.
3. OBTAINED RESULTS

The results of the metrics used to compare the performances of the proposed model are given in Table XV.

| Technique          | DR  | FPR | Precision | f-measure | Accuracy |
|--------------------|-----|-----|-----------|-----------|----------|
| SABADT             | 99.1| 0.02| 99.11     | 99.2      | 99.17    |
| SABADT+10 fold cross val. | 98.3| 0.02| 98.32     | 98.4      | 98.38    |
| Naive Bayes        | 76.1| 0.2 | 78.7      | 76.0      | 76.13    |
| Decision Tree      | 97.0| 0.03| 97.0      | 97.0      | 96.9     |
| Decision Table     | 93.3| 0.06| 93.3      | 93.3      | 93.29    |
| SMO                | 91.6| 0.08| 91.7      | 91.6      | 91.63    |
| AdaBoost           | 88.6| 0.12| 88.9      | 88.5      | 88.56    |

As a result of testing the proposed method, an accuracy rate of 99.17% was obtained in detecting attacks. The Decision Table and Decision Tree techniques also gave similar results, but their performance is poor compared to the proposed method in terms of runtime and memory usage. In addition, looking at the results, Naive Bayes and AdaBoost algorithms cannot generate satisfactory results in datasets where the hints of attacks are not obvious.

The contributions, methods, number of features and accuracy rates of some studies on the UNSW-NB15 dataset are given in Table XVI. Considering the methods in the literature, there are not many studies that develop algorithms for both feature selection and attack detection. Compared to existing ones, Moustafa and Slay’s work is close to the proposed method in terms of features used, but far behind in accuracy. However, Moustafa et al. is close to the proposed method in terms of accuracy, but in this study, there is no study in the area of feature selection, all features are used and standard methods are used for classification.

As seen in Figure 6, only 12% of the literature studies performed on KDD’99 and the UNSW-NB15 datasets, which contributed to both feature selection and classification, 24% of them performed only in one of the mentioned dataset. The remaining part has tested in both datasets, but used existing methods in the literature in at least one of the dataset.

| Paper                | Feature Selection | Classification Technique | Number of used features | Accuracy Rate (%) |
|---------------------|-------------------|--------------------------|-------------------------|-------------------|
| Moustafa and Slay 2015 [54] | Hybrid approach (extracting unnecessary features) | Expectation Maximization (EM) clustering, Naive Bayes (NB) and Logistic Regression (LR) | 11 | 77.2-83 |
| Bhamare et al. 2016 [55] | - | Logistic Regression | 43 | 89.26 |
| Baig et al. 2017 [56] | - | boosting based ANN AdaBoost | 43 | 86.4 |
| Belouch et al. 2017 [57] | Ranker algorithm | RepTree | 20 | 88.95 |
| Anwer et al. 2018 [58] | Filter and Wrapper | J48 and Naïve Bayes | 18 | 88 |
| Aravind and Kalaiselvi 2017 [59] | - | Ensemble classifier | - | 90 |
| Idhammad et al. 2017 [60] | Relevant features | FNN ADDM | - | 97.1 |
| Nguyen et al. 2018 [61] | PCA | Deep Learning | - | 95.84 |
| Moustafa et al. 2018 [62] | - | AdaBoost, DT, NB, ANN | - | 99.54 |
| Beloucha et al. 2018 [63] | - | NB, DT, RF, SVM | 43 | 97.49 |
| Hassine et al. 2019 [64] | Importance-based feature selection | Random Forest | 18 | 97.24 |
| Rajagopal et al. 2020 [65] | selection of relevant features | Neural networks | 23 | 97 |
| **Proposed Technique** | **FSAP** | **SABADT** | **13** | **99.17** |
IV. CONCLUSION
Due to the intense interest in wireless local area networks, research in the field of IDSs have increased. However, there is a lack of a WLAN intrusion detection system that can reliably and accurately detect all possible attacks. A type of attack previously identified as an anomaly may behave differently and cannot be identified. As a result, most techniques used in today’s IDS cannot cope with the dynamic and complex nature of cyber-attacks. In order to solve these problems, a new methodology has been proposed, which contributes to both feature selection and classification processes. Firstly, the Feature Selection Approach (FSAP) which is the basic step of creating a model and consists of developed equation and algorithm is proposed. Secondly, a hybrid SABADT technique which combines signature- and anomaly-based methods are proposed for training and testing.

In order to evaluate the performance of the proposed methodology, the KDD’99 and UNSW-NB15 dataset were used. The obtained results are compared with the standard machine learning techniques results that are mostly used in the literature. As a result of the testing of the proposed method, an accuracy rate of 99.65% and 99.17% were achieved at detecting attacks. In addition, it is aimed to test the proposed method on current attacks within the scope of the study. For this purpose, common hacking tools (Hulk, Nmap, Tor’s Hammer, etc.) were used to obtain a mix of normal activities and current attack behaviors. These tools have been used to perform attacks such as sniffing, overflow, jamming, password guessing, imap, portscan, ipcan, etc. These different types of attacks were captured with the Wireshark tool. Some of the captured attacks were used only in the testing phase. Here, the attacks were detected with an accuracy rate of 99.69%. When comparing the obtained results with the known machine learning techniques, the ranging from 0.6% to 11% better results were obtained according to the detection rate metrics. While testing the model created with the extracted signatures and rules in the proposed study, test datasets containing attack types that are not included in the training datasets were used. Considering the obtained accuracy rates, it is seen that these unknown attack types are also detected with high accuracy. In addition, the time and memory usage of the proposed methodology performed better than other machine learning algorithms. On the other hand, when compared with the studies in the literature, there are very few studies that proposed both feature selection and classification approach. The proposed study gives better results than other studies in terms of the number of used features and accuracy rate. On the other hand, in the testing phase of proposed methodologies in other researches, generally used a single dataset. In this context, our proposed study conducts experiments on more data sets. At the same time, a small data set was created using attack tools, and a study was conducted on it. With these aforementioned aspects, our study provides more important contributions than the studies in the literature. In the future work, it is aimed to determine the types of attacks in addition to determining the attacks by implementing novel approaches. In this process, the accuracy rate can be increased further by reducing the number of used features.

ACKNOWLEDGMENTS
This study is supported under the 2211-E National Scholarship Programme for Ph.D. student.

REFERENCES
[1] E. Deniz and R. Samet. A new model for secure joining to ZigBee 3.0 networks in the internet of things. In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBGDELFIT) (pp. 102-106). IEEE, 2018, December.
[2] Ö. Aslan, R. Samet and Ö.O. Tannöver. Using a Subtractive Center Behavioral Model to Detect Malware. Security and Communication Networks, 2020.
[3] Ö. Aslan and R. Samet. A comprehensive review on malware detection approaches. IEEE Access, 8, 6249-6271, 2020.
[4] L. Xiao and Y. Liu. A two-step feature selection algorithm adapting to intrusion detection. In 2009 International Joint Conference on Artificial Intelligence (pp. 618-622). IEEE, 2009, April.
[5] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac and P. Faruki. Network intrusion detection for IoT security based on learning techniques. IEEE Communications Surveys & Tutorials, 21(3), 2671-2701, 2019.
[6] K. Ethala, R. Seshadri, N.G. Renganathan and M.S. Saravanan. A ROLE OF INTRUSION DETECTION SYSTEM FOR WIRELESS LAN USING VARIOUS SCHEMES AND RELATED ISSUES. American Journal of Applied Sciences, 10(9), 979, 2013.
[7] L. Pan and X. Xie. Network Intrusion Detection Model Based on PCA+ ADASYN and XGBoost. In Proceedings of the 2020 3rd International Conference on E-Business, Information Management and Computer Science (pp. 44-48), 2020, December.
[8] R.K. Ganesh, R.V. Purushothama, S. Bhimavaram and A. Pradesh. An Effective Analysis on Intrusion Detection Systems in Wireless Mesh Networks. International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017.
[9] R. Mitchell and R. Chen. A survey of intrusion detection in wireless network applications. Computer Communications, 42, 1-23, 2014.
[10] M., Ozkan-Okay, and R. Samet. Hybrid Intrusion Detection Approach for Wireless Local Area Network. In 2020 7th International Conference on Control and Optimization with Industrial Applications (COIA) (pp. 311-313), 2020.
[11] Ö. Aslan, M., Ozkan-Okay and D. Gupta. A Review of Cloud-Based Malware Detection System: Opportunities, Advances and Challenges. European Journal of Engineering and Technology Research, 6(3), 1-8, 2021.
[12] N. Mostafa and I. Slay. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1-3), 18-31, 2016.
[13] M.Ring, S. Wunderlich, D. Scheuring, D. Landes and A. Hotho. A survey of network-based intrusion detection data sets. Computers & Security, 86, 147-167, 2019.
[14] M.G. Cordero, E. Vasilionotakis, A. Wainakh, M. Mühlhäuser and S. Nadjiem-Tehrani. On generating network traffic datasets with synthetic attacks for intrusion detection. ACM Transactions on Privacy and Security (TOPS), 24(2), 1-39, 2021.
[15] A. Singhrova. A host based intrusion detection system for DDoS attack in WLAN. In 2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011) (pp. 433-438). IEEE, 2011.
[16] U.S.K.P.M.Thanthrige, J. Samarabandu and X. Wang. Machine learning techniques for intrusion detection on public dataset. In 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-4), IEEE, 2016.
[17] I. Manzoor and N. Kumar. A feature reduced intrusion detection system using ANN classifier. Expert Systems with Applications, 88, 249-257, 2017.

VOLUME XX, 2017 16
[18] M. Usha and P. Kavitha. Anomaly based intrusion detection for 802.11 networks with optimal features using SVM classifier. Wireless Networks, 23(8), 2431-2446, 2017.

[19] R. Primathra and B.A. Tama. Anomaly detection using random forest: A performance revisited. In 2017 International conference on data and software engineering (CoDSE) (pp. 1-6). IEEE, 2017.

[20] R. Abdulhammed, M. Faezipour, A. Abuzneid & A. Alessa. Effective features selection and machine learning classifiers for improved wireless intrusion detection. In 2018 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-6). IEEE, 2018.

[21] S. Zwan, P. Tarwifyeri and M. Adiguin. Performance Analysis of Machine Learning Classifiers for Intrusion Detection. In 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC) (pp. 1-5). IEEE, 2018.

[22] R. Magán-Carrion, D. Urda, I. Diaz-Cano and B. Dorronsoro. Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. Applied Sciences, 10(5), p. 1775, 2020.

[23] M. Lopez-Martín, B. Carro and A. Sanchez-Esguevillas. Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Systems with Applications, 141, 112963, 2020.

[24] T. Duan, Y. Tian, H. Zhang, Y. Liu, Q. Li, J. Jiang and Z. Shi. Intelligent Processing of Intrusion Detection Data. IEEE Access, 8, 78330-78342, 2020.

[25] S.M. Kasongo and Y. Sun. A deep learning method with wrapper based feature extraction for wireless intrusion detection system.

[26] Computers & Security, 92, 101752, 2020.

[27] A. Shaihtab, R. Moskovitch, Y. Eliovici and C. Glezer. Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey. information security technical report, 14(1), 16-29, 2009.

[28] D. Başkaya and R. Samet. DDoS Attacks Detection by Using Machine Learning Methods on Online Systems. In 2020 5th International Conference on Computer Science and Engineering (UBMK) (pp. 52-57). IEEE, 2020, September.

[29] G. Pu, L. Wang, J. Shen and F. Dong. A hybrid unsupervised clustering-based anomaly detection method. Tsinghua Science and Technology, 26(2), 146-153, 2020.

[30] R. Mohanabharathi, M.T. Kalaikumaran and S. Karthi. Feature selection for wireless intrusion detection system using filter and wrapper model. International Journal of Modern Engineering Research (IJMER), 2(4), 1552-1556, 2012.

[31] O. El-Farraj, M. Moskovich, N. Araújo, R. de Oliveira, A.A. Shinoda and B. Bhargava. Signature-based Denial of Service and Probe Detection, a Machine Learning approach. International Journal of Computer Applications, 78(4), 2013.

[32] L. Arockiam, B. Vani, S. Sivagowry and A. Persia. “A solution to prevent resource flooding attacks in 802.11 WLAN”. In: Global Trends in Computing and Communication Systems. Springer, pp. 607–616, 2012.

[33] M. Malezkadeh, A.A. Ghani, J. Desa and S. Subramanian. “Empirical analysis of virtual carrier sense flooding attacks over wireless local area network.” In: Journal of Computer science 5.3, p. 214, 2009.

[34] A. Martínez, U. Zurutuza, R.Urbieetxeberia, M. Fernández, J. Lizarraga, A. Serna and I. Vélez. “Beacon Frame Spoofing Attack Detection in IEEE 802.11 Networks”. In: Availability, Reliability and Security. ARoS 08. Third International Conference on IEEE, pp. 520–525, 2008.

[35] A.M. Patel, A.R. Patel and H.R. Patel. Rap problems and solutions in 802.11 wireless LAN. International Journal, 21(1), 2014.

[36] H. Gonzales, K. Bauer, J. Lindqvist, D. McCoy and D. Sicker. “Practical defenses for evil twin attacks in 802.11”. In: Global Telecommunications Conference (GLOBECOM), IEEE, pp. 1–6, 2010.

[37] A. Klein. Attacks on the RC4 stream cipher. Designs, codes and cryptography, 48(3), 269-286, 2008.

[38] M.J. Gowadiya and A. Jain. Intrusion Detection in KDD99 Dataset: A Review. International Journal of Scientific & Engineering Research, 6(11), 2015.

[39] C. Kollas, G. Kambourakis, A. Stavrou and S. Grtilzalis. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials, 18(1), 184-208, 2015.

[40] S.J. Yu, P. Koh, H. Kwon, D.S.Kim and H. K. Kim. Hurst parameter based anomaly detection for intrusion detection system. In 2016 IEEE International Conference on Computer and Information Technology (CIT) (pp. 234-240). IEEE, 2016, December.

[41] V. Kumar, D. Sinha, A.K. Das, S.C. Pandey and R.T. Goswami. An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. Cluster Computing, 23(2), 1397-1418, 2020.

[42] S.S. Sawwashere and S.U. Nimbhorkar. Survey of RTS-CTS attacks in wireless network. In 2014 Fourth International Conference on Communication Systems and Network Technologies (pp. 752-755). IEEE, 2014, April.

[43] K. Ghanem, F.J. Aparicio-Navarro, K.G. Kyriakopoulos, S. Lambotharan and J.A. Chambers. Support vector machine for network intrusion and cyber-attack detection. In 2017 Sensor Signal Processing for Defence Conference (SSPD) (pp. 1-5). IEEE, 2017.

[44] N. Araújo, R. de Oliveira, A.A. Shinoda and B. Bhargava. Identifying important characteristics in the KDD99 intrusion detection dataset by feature selection using a hybrid approach. In 2016 17th International Conference on Telecommunications (pp. 552-558). IEEE, 2016, April.

[45] F. Amiri, M.R. Yousefi, C. Lucas, A. Shakeri and N. Yazdani. Mutual information-based feature selection for intrusion detection systems. Journal of Network and Computer Applications, 34(4), 1184-1199, 2011.

[46] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai and K. Dai. An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert systems with applications, 39(1), 424-430, 2012.

[47] Z. Karimi, M.M.R. Kashani and A. Harounabadi. Feature ranking in intrusion detection dataset using combination of filtering methods. International Journal of Computer Applications, 78(4), 2013.

[48] H. Saxena and V. Richarya. Intrusion detection in KDD99 dataset using SVM-PSO and feature reduction with information gain. International Journal of Computer Applications, 98(6), 2014.

[49] A.A. Olusola, A.S. Oladele and D.O Abosede. Analysis of KDD’99 intrusion detection dataset for selection of relevance features. In Proceedings of the world congress on engineering and computer science (Vol. 1, pp. 20-22). WCCECS, 2010, October.

[50] M.A.M. Hasan, M. Nasser, S. Ahmad and K.I. Molla. Feature selection for intrusion detection using random forest. Journal of information security, 7(3), 129-140, 2016.

[51] C. Babirye and E. Mwebaze. Signature-based Denial of Service and Probe Detection, a Machine Learning approach. International Journal of Technology and Management, 3(2), 11-11, 2018.

[52] S. Kaur and M. Singh. Hybrid intrusion detection and signature generation using Deep Recurrent Neural Networks. Neural Computing and Applications, 1-19, 2019.

[53] C.Iwendi, S. Khan, J.H. Anajemba, M. Mittal, M. Alenezi and M. Alazab. The use of ensemble models for multiple class and binary problems. Expert Systems with Applications, 141, 112963, 2019.

[54] M. M. Baig, M. M. Awais and, E.S. M. El-Feky, ”arXiv preprint arXiv:1707.05505, 2015.

[55] D. Bhamare, T. Salman, M. Samaka, A. Erbad and R. Jain, “Feasibility of supervised machine learning for cloud security.” In Proceedings of the 6th International Conference on Information Science and Security (ICISS) (pp. 1-5). IEEE, 2016, December.

[56] M. M. Baig, M. M. Awais and, E.S. M. El-Feky, ”arXiv preprint arXiv:1707.05505, 2015.

[57] D. Bhame, T. Salman, M. Samaka, A. Erbad and R. Jain, “Feasibility of supervised machine learning for cloud security.” In Proceedings of the 6th International Conference on Information Science and Security (ICISS) (pp. 1-5). IEEE, 2016, December.

[58] M. M. Baig, M. M. Awais and, E.S. M. El-Feky, ”arXiv preprint arXiv:1707.05505, 2015.

[59] D. Bhame, T. Salman, M. Samaka, A. Erbad and R. Jain, “Feasibility of supervised machine learning for cloud security.” In Proceedings of the 6th International Conference on Information Science and Security (ICISS) (pp. 1-5). IEEE, 2016, December.
[57] M. Belouch, S. El Hadaj and M. Idhammad, “A two-stage classifier approach using reptree algorithm for network intrusion detection,” International Journal of Advanced Computer Science and Applications, 8(6), 389-394, 2017.

[58] H. M. Anwer, M. Farouk and A. Abdel-Hamid, “A framework for efficient network anomaly intrusion detection with features selection,” In 2018 9th International Conference on Information and Communication Systems (ICICS) (pp. 157-162). IEEE, 2018, April.

[59] M. M. Aravind and V.K.G. Kalaiselvi, “Design of an intrusion detection system based on distance feature using ensemble classifier,” In 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN) (pp. 1-6). IEEE, 2017, March.

[60] M. Idhammad, K. Afdel and M. Belouch, “Dos detection method based on artificial neural networks,” International Journal of Advanced Computer Science and Applications, 8(4), 465-471, 2017.

[61] K.K. Nguyen, D.T. Hoang, D. Niyato, P. Wang, D. Nguyen and E. Dutkiewicz, “Cyberattack detection in mobile cloud computing: A deep learning approach,” In 2018 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-6). IEEE, 2018, April.

[62] N. Moustafa, B. Turnbull and K.K.R. Choo, “An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things,” IEEE Internet of Things Journal, 6(3), 4815-4830, 2018.

[63] M. Belouch, S. El Hadaj and M. Idhammad, “Performance evaluation of intrusion detection based on machine learning using Apache Spark,” Procedia Computer Science, 127, 1-6, 2018.

[64] K. Hassine, A. Erbad and R. Hamila, “Important complexity reduction of random forest in multi-classification problem,” In 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC) (pp. 226-231). IEEE, 2019, June.

[65] S. Rajagopal, K.S. Hareesha and P.P. Kundapur, “Feature Relevance Analysis and Feature Reduction of UNSW NB-15 Using Neural Networks on MAMLS,” In Advanced Computing and Intelligent Engineering (pp. 321-332). Springer, Singapore, 2020.

Merve Ozkan-Okay received B.S. and M.S. degree in computer engineering from Ankara University, in 2014 and 2016 respectively. She is a Research Assistant and doing Ph.D. in Department of Computer Engineering, Ankara University. Her current research interests include cyber security, cloud-based systems, machine learning and image processing. She has published several papers on international journals and conferences.

Ömer Aslan is a Dr. researcher in the Department of Information Technologies at the University of Bandırma 17 Eylül, Turkey. He received his PhD in cyber security field 2020 from university of Ankara, Turkey, MSc in information security field in 2014 from university of Texas at San Antonio, United States of America (USA), and BSc in computer engineering department in 2009 at university of Trakya, Turkey. He is working on computer systems, information security, cyber security, malware analysis and cloud computing. He has published several papers on international journals and conferences.

Recep Eryigit – Associate Professor at Ankara University Computer Engineering Department. He got his PhD thesis at Hacettepe University. His research interests in Software Engineering, Operating Systems, Database Management and Quantum Algorithms. He has published several papers on international journals and conferences.

Refik Samet - Professor and Head of Department at Ankara University Computer Engineering Department. He got his PhD thesis in Russian Academy of Sciences, Institute of Control Sciences Department of Fault-Tolerant Multicomputer and Multiprocessor Systems. His research interests include Parallel Systems, Cyber Security, Machine Learning Network and Mobile Applications. He has published several papers on international journals and conferences.

VOLUME XX, 2017

16