DOI: 10.24427/978-83-66391-87-1_05

Olexander BELEJ[1], Nataliia BOKLA[1], Tadeusz WIĘCKOWSKI[2]

5. DEVELOPMENT OF AN ALGORITHM FOR DETECTING ATTACKS IN SENSOR WIRELESS SYSTEMS

Wireless networks have gained immense popularity. Their widespread distribution is due to undeniable advantages over traditional cable networks: ease of deployment, user mobility in the network coverage area, easy connection of new users. On the other hand, the security of such networks often limits their application. If an attacker needs to have a physical connection to the network when attacking a wired network, then in the case of wireless networks, he can be anywhere in the network coverage area. Also, these networks are subject, including due to protocol imperfections, to specific attacks, which will be discussed below. On the other hand, the low level of security of such networks often limits their application. Also, these networks are subject, including due to protocol imperfections, to specific attacks, which will be discussed below.

In connection with the foregoing, researchers are looking for possible improvements to current protocols. In [1], the author proposes to encrypt the entire MAC protocol data unit (MPDU), including MAC headers, except for the FCS frame check sequence, which, will lead to noticeable delays in data transmission and low channel bandwidth. Another approach is to put in the control frame a hash of a certain string known only to a specific sender, by transmitting which in the future it can be uniquely identified and processed [2]. However, this method only prevents one type of attack.

In practice, to protect against network attacks, ordinary users and small organizations, as a rule, are limited to using anti-virus software, which at the present stage of development has some additional protection modules [3]. Large enterprises are forced to purchase expensive wireless intrusion detection systems (WIDS). However, currently, there are no generally accepted standards in this area; manufacturers use closed algorithms for detecting and classifying attacks. In this case, the task of attributing a fragment of network traffic to some type of attack or normal network activity can be solved by applying the methods of data mining (DM) [4].

[1] Lviv Polytechnic National University, Ukraine
[2] Wroclaw University of Science and Technology, Poland

In [5, 6], to solve this problem, the use of neural networks and the support vector method Support Vector Machine (SVM) are proposed. In [7], an approach to the organization of a neural network attack detection system based on a two-layer perceptron and Kohonen network was considered.

It is worth noting that the above studies relate to the detection of intrusions into traditional wired networks [8]. However, there are no works on the targeted use of DM methods to detect attacks specific to local wireless networks. For this reason, this material discusses the main types of attacks inherent in wireless networks, some recommended methods of protection against them, and also proposes the architecture of an attack detection system based on DM methods and evaluates the effectiveness of the attack detection algorithms used in it.

5.1. ATTACKS IMPLEMENTED IN WIRELESS NETWORKS

The attacks on wireless networks are based on intercepting network traffic from an access point or traffic between two connected stations, as well as introducing additional data into a wireless communication session. To form a better understanding of the types of wireless attacks that an attacker can carry out against a wireless network, it is important to classify them. So, attacks can be aimed at different layers of the OSI model: application, transport, network, channel and physical.

Depending on the purpose of the attack, characteristic of the 802.11 protocol family can be divided into several categories [9]: obtaining unauthorized access to the network; integrity violation; privacy violation; access violation; identity theft.

Depending on the purpose of the attack on local wireless networks, OSI models can be divided into several categories [10]:

- gaining unauthorized access to the network: rogue access point, spoofing MAC, hacking a network client, hacking access points,
- integrity violation: 802.11 frame injection, play 802.11 data, delete 802.11 data, play 802.1X EAP, play 802.1X RADIUS,
- breach of confidentiality: eavesdropping, evil twin, AP phishing, the man in the middle,
- accessibility violation: radiofrequency noise, Queensland DoS, Flood Request Probe, Associate / Authenticate / Disconnect / Deauthenticate Flood, 802.1X EAPStart, EAPFailure Flood,
- authentication Bypass: Pre-Shared Key, Identity Theft 802.1X, 802.1X EAP Downgrade, password cracking 802.1X, hacking domain accounts, hacking WPS PIN.

These attacks are based on the use of vulnerable wireless networks represented in the WVE database [11]:

- sending Probe requests with a zero-length SSID tag field (WVE-2006-0064),
- EAP Logoff attack (WVE-2005-0050),
- RTS / CTS flood (WVE-2005-0051),

- WLAN flooding with dissociation packets (WVE-2005-0046),
- WLAN flooding with deauthentication packets (WVE-2005-0045),
- KARMA wireframe (WVE-2006-0032),
- sending an invalid deauthentication reason code,
- sending too long an SSID (WVE-2006-0071, WVE-2007-0001),
- sending an Airjack beacon frame (WVE-2005-0018),
- sending invalid channel numbers in beacon frames (WVE-2006-0050).

Table 5.1. The ratio of the number of attack signatures in the training base

Normal	67343		
DoS		R2L	
Class	Quantity	Class	Quantity
neptune	41214	guess_passwd	162
smurf	2646	ftp_write	8
Pod	201	imap	11
teardrop	892	phf	4
land	18	multihop	7
back	956	warezmaster	40
U2R		Probe	
Class	Quantity	Class	Quantity
buffer_overflow	30	portsweep	2931
load-module	9	upsweep	3599
Perl	3	satan	3633
rootkit	10	nmap	1493

Table 5.2. The ratio of the number of attack signatures in the test base

Normal	9711		
DoS		R2L	
Class	Quantity	Class	Quantity
neptune	4657	guess_passwd	1231
smurf	665	ftp_write	3
Pod	41	imap	1
teardrop	12	phf	2
land	7	multihop	18
back	359	warezmaster	944
U2R		Probe	
Class	Quantity	Class	Quantity
buffer_overflow	20	portsweep	157
load-module	2	upsweep	141
Perl	2	satan	735
rootkit	13	nmap	73

Testing the wireless access level for WPA2-Enterprise. A connection is a sequence of packets starting and ending at specific points in time, between which data streams are transferred from the source IP address to the recipient IP address using a specific protocol [12]. Each connection is designated as normal or as some type of attack from four categories of attacks: Denial of Service (DoS), unauthorized obtaining of user rights Remote to Local (R2L), the unauthorized elevation of user rights to superuser User to Root

(U2R) and sensing (Probe). The ratio of the number of attacks of different types is shown in Table 5.1, 5.2.

Some of these types of attacks are the costs of the technology of radiofrequency data transmission, and also depend on the human factor and must be addressed using organizational measures. Wireless intrusion detection (WIDS) systems should be distinguished from network security hardware, except for firewalls.

5.2. ATTACK DETECTION SYSTEM

The decision on the security of any network activity in commercial products is implemented using closed algorithms, the principle of which is a commercial secret. Moreover, the declared number and types of detected attacks for different products differ, although, in reality, they belong to the same type of attacks, which is explained by the lack of standards in the classification.

The tasks of detecting and classifying attacks can be solved by using data mining (DM) methods, which allow revealing significant correlations, patterns, and trends in large amounts of data. The proposed system uses algorithms for constructing a classification model based on the support vector method, the method of k-nearest neighbors, neural networks and decision trees.

The proposed architecture of an intelligent attack detection system has a modular scheme for organizing interaction between components with a dedicated sensor subsystem and centralized management through the administrator console. The architecture of the system is shown in Fig. 5.1.

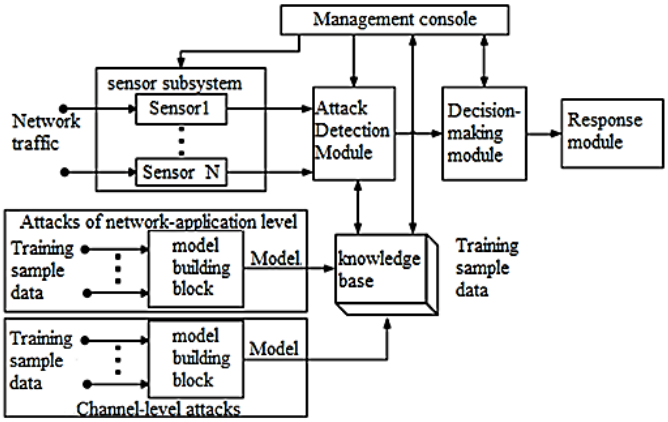

Fig. 5.1. Structure of the Attack Detection System

The basis for identifying attacks is a knowledge base, the construction of which at the stage of the initial configuration of the system provides a block for constructing a classification model. The classification model is built based on the signatures of the training sample and then used to classify real network activity.

The attack detection module of the designed attack detection system can be functionally divided into submodule for detecting attacks of the network, transport and application levels, link-level attack detection submodule.

The system operates in two models:

- configuration model, when a set of signatures is loaded into the block for constructing the classification model as input, each of which is a pair {traffic parameters vector | type of attack},
- normal operation model, when the values of the traffic parameters are supplied as input to the sensor subsystem.

The basis for identifying attacks is a knowledge base, the construction of which at the stage of the initial configuration of the system provides a block for constructing a classification model. The classification model is built based on the signatures of the training sample and then used to decide the security of any network activity. In commercial products, this is implemented using closed algorithms, the principle of which is a trade secret. Moreover, the declared number and types of detected attacks for different products differ, although, in reality, they belong to the same type of attacks, which is explained by the lack of standards in the field of wireless attacks.

As shown in the aforementioned works, the tasks of detecting and classifying attacks can be solved by using DM methods to identify significant correlations, patterns, and trends in large amounts of data.

Next, we consider in more detail the methods of DM, which form the basis of the algorithm for constructing a classifying model of the proposed system.

5.3. METHODS FOR ANALYSIS OF ATTACKS IN SENSOR WIRELESS NETWORKS

The Support Vector Method (SVM) refers to linear classification methods. Each state of the system is represented as a point in a multidimensional space, the coordinates of which are the characteristics of the system. Two sets of points belonging to two different classes are separated by a hyperplane in this space. In this case, the hyperplane is constructed so that the distances from it to the nearest instances of both classes are maximum, which ensures the greatest classification accuracy.

Fig. 5.2 shows an example of classifying objects in two-dimensional space using SVM. The figure shows a training data set, which is a set of points of the form $\{x_i, y_i\}$, $i = 1, \ldots, l$, where $x_i \in R^n, y_i \in \{1, -1\}$ is an indicator of the class to which the point belongs x_i. The classes of points are linearly separable, that is, there is such a hyperplane, on one side of which there are points of the class $y_i = 1$, and on the other of the class $y_i = -1$. Points located directly on the hyperplane satisfy the equation

$$\boldsymbol{\omega} \cdot x - \mathbf{b} = 0, \tag{5.1}$$

where the vector $\boldsymbol{\omega}$ is the perpendicular to the dividing hyperplane, the quantity $|\mathbf{b}|/\|\boldsymbol{\omega}\|$

(the absolute value of **b** divided by the modulus of the vector **ω**) determines the distance from the origin to the hyperplane, the operator "·" denotes the scalar product in the Euclidean space in which the data lies.

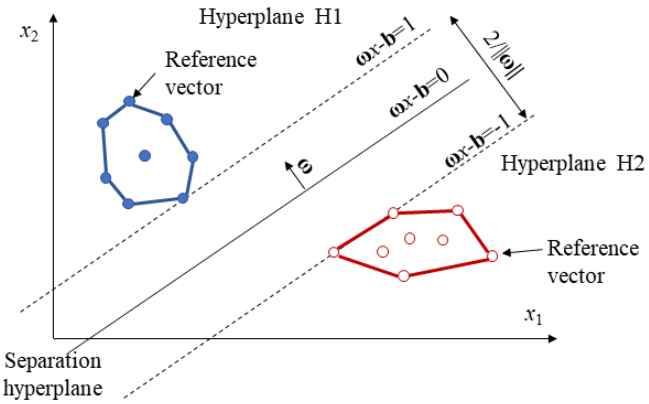

Fig. 5.2. Classification of support vectors

All points for which the condition $\boldsymbol{\omega} \cdot x_i - \mathbf{b} = 1$, lie in the hyperplane H_1 parallel to the separating hyperplane and at a distance $|1 - \mathbf{b}|/\|\boldsymbol{\omega}\|$ from the origin. Similarly, those points for which the condition $\boldsymbol{\omega} \cdot x_i - \mathbf{b} = -1$, lie in the hyperplane H_2 parallel to the plane H_1 and the separating hyperplane, at a distance $|-1 - \mathbf{b}|/\|\boldsymbol{\omega}\|$ from the origin. Thus, the distance between the plane and the positive reference vector is $1/\|\boldsymbol{\omega}\|$, and therefore, the width of the strip is $2/\|\boldsymbol{\omega}\|$.

The advantages of this method are high accuracy, generalization ability, and low computational complexity of decision making. The disadvantage is the relatively large computational complexity of constructing a classifying model.

The method for detecting attacks based on the support vector method is investigated. This was used to construct a classification model from the data of the training sample. The model was tested on attacks such as buffer overflow, rootkit, and SYN flood and showed the relevance of using the support vector method as the basis for an attack detection system.

The *k*-nearest neighbor (*k*-NN) method is a classification method whose basic principle is to assign to the object the class that is most common among the neighbors of this object. Neighbors are formed from many objects whose classes are already known, and, based on a given value of k ($k \geq 1$), it is determined which of the classes is the most numerous among them. If $k = 1$, then the object simply belongs to the class of the only nearest neighbor. The *k*-NN method is one of the simplest DM methods. The disadvantage of the *k*-NN method is that it is sensitive to the local data structure.

Neural networks make it possible to solve practical problems associated with pattern recognition and classification. A neural network consists of interconnected neurons that form the input, intermediate and output layers. Training takes place by adjusting the weights of neurons to minimize classification errors. The advantages of neural networks are their ability to automatically acquire knowledge in the learning process, as well as the ability to generalize. The main disadvantage is sensitivity to noise in the input data.

Decision trees are a tree structure of *leaves* and *branches*. On the edges of the decision, the tree is written the attributes that the objective function depends on, the values of the objective function are written in the *leaves*, and the attributes that distinguish the objects are written in the other nodes. To classify a new object, you need to go down the tree from the root to the leaf and get the corresponding class, the path from the root to the leaf acts as classification rules based on the values of the attributes of the object.

The advantages of decision trees are the simple principle of their construction, good interpretability of the results, the disadvantage is the low accuracy of classification.

Further, to identify the most effective method for constructing a classification model with a wireless attack detection system, a comparison of the considered DM methods will be given.

5.4. ANALYSIS OF CYBER ATTACKS IN SENSOR WIRELESS SYSTEMS

The recognition accuracy of the considered types of attacks using SWS was assessed by comparing the classification results using various DM methods. Based on the above classification of attacks by OSI model levels, attacks on local wireless networks are divided into two groups:

- physical and link-layer attacks that are specific to wireless networks;
- attacks from the network to the application layers inherent in any technology for organizing local area networks, including Ethernet.

The corresponding submodule of attack detection of the proposed system during the experiments uses the signatures of the NSL KDD-2009 base as an example of a network and application-layer attacks. To form a training sample of wireless attacks of the channel and network levels, a test local wireless network with access protection technology WPA2-PSK was organized. The collected packages were analyzed and reduced to the form used in the NSL-KDD-2009 database.

Initially, 41 attributes were used to describe the attacks in the NSL-KDD-2009 database, which reflect the application, transport, and network layers of the OSI model. The selected features are presented in Table 5.3. To describe attacks characterized by a large number of connections to the destination node, a window lasting two seconds was selected (DoS attacks), as well as a window of 100 connections to the same node (Probe).

The first step was the processing of data from the database since for the error-free operation of the algorithms, all attributes must have numerical values distributed between zero and one. For this, text attributes were converted to binary, while numerical ones were normalized concerning the minimum and maximum values.

After that, the data of the training sample were fed to the input of the building block of the classifying model, which forms the basis of the knowledge base, by various DM methods. Then, the attack detection module classified the records of the test set based on the corresponding model according to the criteria contained in the knowledge base and assigned a label to the class of network activity.

Table 5.3. Important traffic settings for network and application layers

Features	Description	Type
Characteristics of the TCP compound		
duration	Connection time (s)	numerical
protocol_type	Transport layer protocol	text
service	Application layer service	text
flag	Status of connection	binary
src_bytes	Incoming stream, byte	numerical
dst_bytes	Outbound stream, byte	numerical
land	The addresses are the same, 0 otherwise	binary
wrong_fragment	Number of incorrect fragments	numerical
urgent	Number of urgent packages	numerical
Session Features		
hot	Number of *hot* indicators	numerical
num_failed_logins	Number of failed login attempts	numerical
logged_in	Successful entry	binary
root_shell	Access with administrative credentials	binary
num_root	Number of access attempts with administrative credentials	numerical
num_shells	Number of attempts to use the command line	numerical
num_access_files	Number of operations with access control files	numerical
Stats in 2 seconds / 100 connections		
count / dst_host_count	Number of connections with a matching host	numerical
serror_rate/ dst_host_serror_rate	% connection with error SYN	numerical
rerror_rate / dst_host_same_src_port_rate	% connections with REJ error /% connections with the same source port	numerical
same_srv_rate / dst_host_same_srv_rate	% of connections with the same service	numerical
diff_srv_rate / dst_host_diff_srv_rate	% connection to various services	numerical
srv_count / dst_host_srv_count	Number of connections with matching service	numerical
srv_serror_rate / dst_host_srv_serror_rate	% connections with SYN error	numerical
srv_rerror_rate / dst_host_srv_rerror_rate	% connections with error REJ	numerical
srv_diff_host_rate / dst_host_srv_diff_host_rate	% connections with different hosts	numerical

Based on the coincidence of the estimated and actual class labels, the effectiveness of attack detection was evaluated by the following criteria.

1. The total percentage of correctly classified attacks A (accuracy)

$$A = \frac{TP + TN}{N}, \tag{5.2}$$

where TP is the number of true-positive records, TN is the number of true-negative records, N is the total number of classified records.

2. The accuracy of the classification P (precision):

$$P = \frac{TP}{TP + FP}, \tag{5.3}$$

where FP is the number of false-positive records.

3. Completeness of classification R (recall):

$$R = \frac{TP}{TP + FN}, \tag{5.4}$$

where FN is the number of false-negative entries.

The traffic parameters used to describe the data link attack signatures are shown in Table 5.4. The experiments were carried out according to the algorithm shown in Fig. 5.3.

Table 5.4. Important traffic settings for network and application layers

Features	Description	Type
802.11 Protocol Features		
frame_ type/subtype	Frame Type / Subtype	text
protocol_type	Link Protocol Type	text
source_address	Source MAC Address	text
destination_address	Destination MAC address	text
Length	Frame size, bytes	numerical
SSID	SSID tag value	text
sequence_number	Frame number	numerical
fragment_number	Fragment Number	numerical
DS_status	Distributed system sharing	numerical
more_fragments	More fragments for transmission, 0 otherwise	binary
retry	Retransmission of the previous frame, 0 otherwise	binary
pwr_mgt	The client is in power saving mode, 0 otherwise	binary
more_data	Buffered frames for transmission, 0 otherwise	binary
protected_flag	Frame data is encrypted, 0 otherwise	binary
order_flag	Processing frames strictly in order, 0 otherwise	binary
duration	ACK + SIFS Transmission Duration, μs	numerical
chan_number	Channel number	numerical
signal	The signal level of the transmitter, %	numerical
TX_rate	Baud Rate, Mbps	numerical
cipher	Used encryption algorithm	textual
reason_code	Deauthentication Reason Code	numerical
Statistics in 2 seconds		
mng_frm_count	The number of management personnel	numerical
ctrl_frm_count	The number of control frames	numerical
probe_count	Number of connection requests	numerical
frag_count	The average number of fragmented packets	numerical

The support vector method was implemented using the SVS C-SVC library LibSVM, and the radial basis function (RBF) was used as the kernel function. The maximum learning error was limited to 10^{-5}.

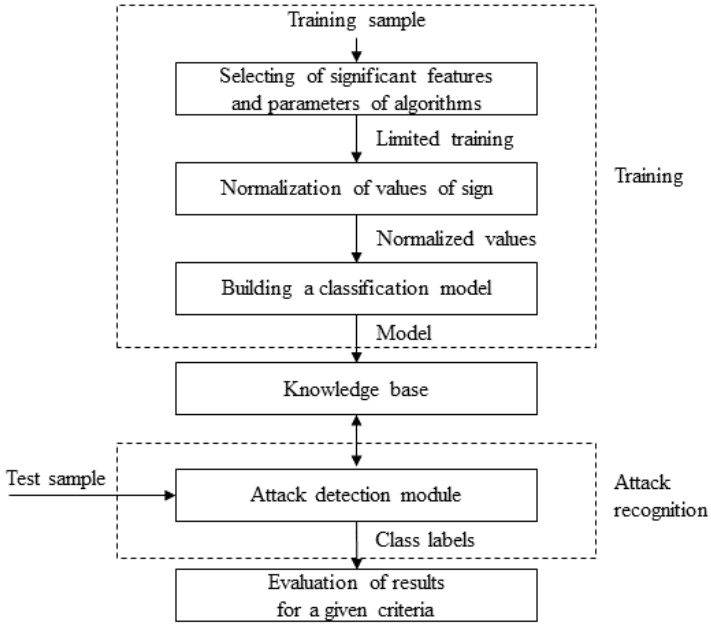

Fig. 5.3. Algorithm for attack detection in sensorless systems

The classification results using various DM methods are shown in Tables 5.5 and 5.6.

When classified by the method of *k*-nearest neighbors experimentally, as the optimal parameters of the algorithm, we chose a value of *k* equal to five, and the metric is the Manhattan distance.

The neural network was implemented as a multilayer perceptron with two hidden layers. Training with duration of 1500 cycles was carried out using the back propagation algorithm of the error. The maximum learning error was 10^{-7}.

Table 5.5. Network Application Layer Attack Performance Indicators

Group	Network activity class	Support Vector Method		k-nearest neighbors		Neural network		Decision trees	
		fullness	accuracy	fullness	accuracy	fullness	accuracy	fullness	accuracy
DoS	neptune	98.97	99.98	97.25	97.50	99.36	99.98	97.32	99.93
normal	normal	96.56	92.28	96.55	93.63	97.07	87.25	97.10	90.98
R2L	guess_passwd	76.69	100	66.86	95.48	66.37	97.03	65.72	99.88
DoS	smurf	100	99.70	97.59	100	95.19	99.53	100	100
Probe	satan	93.74	76.47	94.83	76.76	90.75	81.84	96.19	80.62
U2R	buffer_overflow	25.00	62.50	35.00	100	0	0	25.00	62.50
DoS	back	98.05	98.60	99.44	100	96.10	97.73	77.16	92.33
R2L	warezmaster	59.11	99.11	82.20	99.74	16.10	98.06	63.56	100
DoS	pod	95.12	72.22	95.12	72.22	82.93	70.83	95.12	46.99
Probe	nmap	98.63	93.51	97.26	91.03	79.45	90.62	98.63	74.23
Probe	ipsweep	97.16	93.84	97.16	74.86	97.87	79.31	99.29	88.05
probe	portsweep	91.08	56.30	85.35	73.22	89.17	61.67	84.71	54.07
DoS	teardrop	83.33	21.28	83.33	14.08	75.00	18.75	100	24.49
DoS	land	57.14	100	57.14	100	0	0	14.29	100
Average		83,61	83.27	84.65	84.89	70.38	70.19	79.58	79.58

Table 5.6. Link Level Attack Performance Indicators (in %)

Class	Support Vector Method		k-nearest neighbors		Neural network		Decision trees	
	fullness	accuracy	fullness	accuracy	fullness	accuracy	fullness	accuracy
Normal	98.03	92.49	97.65	99.26	94.37	99.38	95.48	95.11
rogue_client	100	37.56	6.22	20	32.44	20	100	69.02
EAPOL_logoff_flood	8.82	100	26.85	100	0.12	100	44.08	100
auth_flood	85.14	94.03	100	93.67	100	92.50	97.30	100
EAPOL_start_flood	100	100	100	50.58	100	44.14	100	100
deauth_flood	100	99.10	100	99.75	100	84.39	100	100
caffe_latte	0	0	100	100	100	70.97	100	100
Chopchop	100	62.86	100	100	100	3.28	100	2.27
client_fragment	97.44	99.77	100	99.89	100	96.98	100	100
AP_fragment	98.73	97.01	99.75	98.25	100	98.26	100	100
data_replay	99.82	98.13	100	99.98	99.96	99.53	100	100
MAC_spoofing	100	6.63	100	10.91	0	0	0	0
evil_twin_AP	100	100	100	64.78	100	94.30	100	94.90
EAP_replay	100	100	100	100	100	100	100	100
beacon_flood	100	100	100	99.95	99.91	100	100	99.86
RTS/CTS_flood	99.82	99.82	100	84.64	100	91.49	100	91.68
fake_auth	55.56	100	66.67	85.71	77.78	10.45	100	100
Average	84.90	81.61	88.07	82.79	82.62	70.92	90.40	85.46

Decision trees were built using the standard operator of the RapidMiner environment, the minimum threshold for the formation of a new node was chosen to be 4, the minimum number of leaves of the node was one, and the maximum number of levels was 10.

As can be seen from Table 5.5, the methods of support vectors and k-nearest neighbors showed close results in the course of detecting attacks, the decision tree and the neural network performed slightly worse. The low percentage of detection of certain types of attacks, such as warezmaster, guess_passwd, buffer_overflow, and land, is caused by the uneven quantitative distribution of training samples for different classes - the predominance of normal signatures and attacks of the DoS and Probe categories. For the same reason, some of the attacks were classified incorrectly, so their results are not presented in Table 5.5. However, according to the indicators in Table 5.6, the *k*-nearest-neighbor method and decision tree are superior to SVM and neural networks in solving the task of detecting link-layer attacks. Thus, the analysis of experimental data shows that the algorithms used to demonstrate different values of attack detection performance indicators depending on the type of network activity and the level of the OSI model at which the attack is implemented.

In this regard, it is proposed to use an ensemble of four developed algorithms and one arbiter, which determines the final class of network activity by weighted voting. The architecture and functioning principles of the proposed ensemble will be the essence of further research.

5.5. CONCLUSIONS

This material provides an overview of network attacks that are relevant for local wireless networks, presents the architecture of the proposed attack detection system based on the use of DM methods for recognizing attack data, and compares these methods during experiments to detect the considered types of attacks.

In general, the methods showed high accuracy and completeness of detection during the experiments, from which it can be concluded that the proposed approach to detecting attacks in local wireless networks is practical.

REFERENCES

[1] OLUSOLA A., OLADELE A., ABOSEDE D., Analysis of KDD'99 Intrusion Detection Dataset for Selection of Relevance Features, in: *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, vol. 1, p. 162-168, 2010.

[2] NGUYEN T., NGUYEN B., PHAM H., An efficient solution for preventing Dis'ing attack on 802.11 networks, in: *2012 International Conference on Green Technology and Sustainable Development: Journal of Engineering Technology and Education*, Hochiminh, p. 395-403, 2012.

[3] BELEJ O. et al., Features of application of data transmission protocols in wireless networks of sensors, in: *Advanced information and communication technologies, AICT–2019: proceedings of the 3rd International Conference*, Lviv, Ukraine, p. 317–322, 2019.

[4] MULAY S., DEVALE P., GARJE G., Intrusion Detection System using Support Vector Machine and Decision Tree, *International Journal of Computer Applications*, vol. 3, no. 3, p. 40-43, 2010.

[5] SUN T., ZHANG J., YANG Y., Review on the development and future trend of the intrusion detection system (IDS), in: *2016 International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, p. 1-6, 2016.

[6] AHMED M. R., CUI H., HUANG X., Smart integration of cloud computing and MCMC based secured WSN to monitor the environment, in: *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE),* Aalborg, p. 1-5, 2014.

[7] HAN W. et al., Quantitative Assessment of Wireless Connected Intelligent Robot Swarms Network Security Situation, *IEEE Access,* vol. 7, p. 134293-134300, 2019.

[8] DONGARE S. P., MANGRULKAR R. S., Implementing energy-efficient technique for defense against Gray-Hole and Black-Hole attacks in wireless sensor networks, in: *2015 International Conference on Advances in Computer Engineering and Applications,* Ghaziabad, p. 167-173, 2015.

[9] ALSHEIKH M.A. et al., Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications, *IEEE Communications Surveys & Tutorials,* vol. 16, no. 4, p. 1996-2018, 2014.

[10] EL MOURABIT Y. et al., Intrusion detection system in Wireless Sensor Network based on mobile agent, in: *2014 Second World Conference on Complex Systems (WCCS),* Agadir, p. 248-251, 2014.

[11] SREERAM I., VUPPALA V.P.K., HTTP flood attack detection in application layer using machine learning metrics and bio-inspired bat algorithm, *Applied Computing and Informatics*, vol. 15, p. 1-5, 2019.

[12] NANDITA S. et. al., Designing of an online intrusion detection system using rough set theory and Q-learning algorithm, *Neurocomputing*, vol. 111, p. 161-168, 2013.