Detection of intrusions based on text analysis and machine learning methods in the development of information systems

Svitlana Popereshnyak[1,*,†], Viktor Ovcharenko[2,†], Yuriy Novikov[2,†] and Hennadii Hulak[3,4,†]

[1] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Prospect Beresteiskyi, 03056 Kyiv, Ukraine

[2] Institute of Software Systems of the National Academy of Sciences of Ukraine, 40-5 Akademik Hlushkov ave., 03187 Kyiv, Ukraine

[3] Borys Grinchenko Kyiv Metropolitan University, 18/2 Bulvarno-Kudriavska str., 04053 Kyiv, Ukraine

[4] Institute of Mathematical Machines and Systems Problems of the National Academy of Sciences of Ukraine, 42 Ac. Glushkov ave., 03680 Kyiv, Ukraine

Abstract

The paper analyzes the main concepts of cyber security, and cyber security technologies, investigates the features of the use of artificial intelligence in cyber security, analyzes the applied methods of machine learning, and presents the results of experimental research on the application of machine learning methods in cyber security. In this work, a host intrusion detection system based on the technique of intelligent text analysis will be implemented. The work describes the difficulties that data sources may face, for example, suffering from the method of complex functions. The paper proposes a classification of methods for detecting SQL injection, XSS, and path traversal attacks, and also provides performance measurements for the specified models. Penetration testing methodology was applied. This technique detects vulnerabilities related to the most popular attacks, such as SQL injection (SQLi), cross-site scripting (XSS), and sensitive data disclosure. Security solutions and suggestions were presented that IT administrators can use as a guide to protect the system against cybercriminal threats. Thus, the effectiveness of the proposed system was substantiated by fixing all detected vulnerabilities to achieve basic security standards. A host-based intrusion detection system (HIDS) was developed using text analysis techniques.

Keywords

cybersecurity, attack, smart home, Internet of Things, machine learning, host, cybercrime, risk, threat, software

1. Introduction

Nowadays, information technologies are becoming an integral part of all areas of our lives in cyberspace. With the development of the Internet of Things (IoT) and the progress of the smart home, network security threats arise. Providing protection becomes important in light of the increasing number of connected devices that may be subject to cyberattacks. This highlights the need for research and implementation of cybersecurity measures in IoT networks and smart homes [1, 2]. These measures are critical to protecting the confidentiality, integrity, and security of data and to ensuring the safety of users and their property [3, 4].

Services aimed at safeguarding networks, systems, and confidential data are referred to as cybersecurity measures, which can be deployed through the utilization of HTTP/HTTPS protocols [5].

Research shows that code injection on web pages is gradually increasing and accounts for up to 96.15% of all web attacks in recent years [6]. Additionally, according to WAAR, which focuses on the most common types of injection attacks [7], cross-site scripting (XSS) attacks account for 49.09% of all web attacks, SQL injection attacks account for 28.32%, and Path attacks Traversal—9.82% [8].

Additionally, user input can be used as an injection tool in attacks against web applications. These user inputs are displayed in the HTTP GET request string. In light of this assumption, a malicious request can be considered as one of the main forms of web injection attacks [9]. Therefore, cybersecurity scientists are developing various methods to detect such malicious activities. Thus, such actions appear as requests in web requests using signature and anomaly detection systems [10].

This research work plans to develop a host intrusion detection system, which will be based on the use of text

CPITS-II 2024: Workshop on Cybersecurity Providing in Information and Telecommunication Systems II, October 26, 2024, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

spopereshnyak@gmail.com (S. Popereshnyak);
viktor.ov4arenko@gmail.com (V. Ovcharenko);
ynovikov@gmail.com (Y. Novikov);
hulak@kubg.edu.ua (H. Hulak)

0000-0002-0531-9809 (S. Popereshnyak);
0009-0002-2402-4771 (V. Ovcharenko);
0009-0006-9800-8765 (Y. Novikov);
0000-0001-9131-9233 (H. Hulak)

mining. To do this, a dataset was selected corresponding to a web server log file containing text entries associated with SQL injection, XSS, and path traversal attacks associated with HTTP GET URL requests. It also examines the challenges that data sources can face, including problems associated with the use of complex functions. Four machine learning models were used in the research process: decision tree, multilayer perceptron (MLP), KNeighbors, and support vector machine. The research utilized four distinct machine learning models: decision tree, multilayer perceptron (MLP), support vector machine, and KNeighbors. In addition, the work proposed a classification of methods for detecting SQL injection attacks, XSS, and path traversal, and also measured the performance of these models.

The primary objective of this study is to investigate the potential enhancements in security mechanisms through the utilization of machine learning techniques for the identification of cyber attacks [11–13].

2. Some fields of application

Let's consider one of the fields of application of this research. We will pay special attention to the direction of the development of information systems of the State Border Service of Ukraine and the quality of development of these systems.

The State Border Service of Ukraine operates with a large amount of sensitive information, in particular data on the movement of persons, goods, and vehicles. Information systems that serve such data must have a high level of protection against cyberattacks, intrusions, or malicious actions. Intrusion Detection Systems (IDS), based on text analysis and machine learning, can help detect anomalies or threats early, preventing data or systems from being compromised.

To ensure the high-quality development and functioning of information systems, it is important to integrate automated monitoring and analysis mechanisms. The use of machine learning algorithms will make it possible to create models that learn from a large amount of data about user actions and network logs to detect threats in real-time. This reduces the human factor and increases the efficiency of response to cyberattacks, which is key to ensuring the uninterrupted operation of systems.

The main goal of optimizing the process of developing information systems is to improve the quality of interaction, productivity, and reliability of systems. The inclusion of IDS using text analysis and machine learning in the development process allows not only to increase security but also to improve the quality of the system itself. This ensures resistance to attacks and increases reliability, which is an important criterion for the quality of State Border Service systems.

The methods of text analysis and machine learning can be applied not only to detect intrusions but also to analyze the behavioral patterns of system users. This allows you to optimize the interface and functionality of the systems according to the real needs of users, while at the same time detecting suspicious activities that may indicate unauthorized access attempts.

Since information threats are constantly evolving, classic protection methods may not be sufficient. Machine learning algorithms allow intrusion detection systems to adapt to new threats because they can update their models based on new data. This creates a dynamic system of protection, which increases the overall quality of the border service information systems development process, ensuring their relevance and security for the long term.

Thus, to improve the quality of the development of information systems of the State Border Service of Ukraine (SBSU), it is necessary to take into account the peculiarities of cyber security and the methods of machine learning described in the paper, due to several key aspects:

1. Dynamics of the development process: In the process of developing information systems for SBSU, the quality of security is one of the most important parameters. Information systems must dynamically adapt to new threats that evolve rapidly, especially in the field of cyber security. Implementing machine learning techniques to detect vulnerabilities such as SQL injection or XSS attacks, as described in the paper, can greatly improve this process.
2. Integration of cyber security methods: To achieve high-quality information systems, it is necessary to integrate technologies that detect intrusions and prevent attacks. This is directly related to the methods described in the paper, such as intrusion detection systems (HIDS), which help protect information systems by ensuring stability and reliability during their development.
3. Optimization through artificial intelligence and machine learning: To optimize the quality of development of information systems, you can use artificial intelligence and machine learning algorithms that not only analyze current data but also predict future threats. The paper mentions text mining that can be applied to data analysis during system development.
4. Ensuring high-security standards: An important stage of quality dynamics management is penetration testing and elimination of vulnerabilities, which are discussed in the paper. Methods such as penetration testing can help IT administrators of the State Security Service to timely identify and eliminate security problems, which will improve the quality of information systems at the development stage.

To optimize the quality management of the process of development of information systems of the State Border Service of Ukraine, it is necessary to use intrusion detection methods based on text analysis and machine learning to ensure the security, adaptability, productivity, and reliability of these systems. This makes it possible to create more secure and effective information systems that meet real needs and challenges in the field of cyber security.

3. Key Security Aspects

3.1. The importance of cybersecurity

Cybersecurity strives to achieve common goals such as protecting and preserving information from theft or attack.

There are three main goals: confidentiality, which ensures the protection of sensitive information, integrity, which ensures the reliability of data, and availability, which ensures the transfer of information. The triad of Confidentiality, Integrity, and Availability (CIA) is a key concept and is considered the fundamental principle of all security systems, guiding overall policy and ensuring complete data protection within organizations [14].

Therefore, it is important to define multiple levels of data protection based on these three aspects of the CIA. The main instruments can be classified as indicated in Table 1.

3.2. Security Features

Cybersecurity has become an area of interest for many organizations and companies in the virtual environment. Data-driven technology tools have attracted significant attention due to the threat of data privacy and security violations. Organizational networks are becoming more accessible, and security issues are becoming more pressing. Every organization that uses Internet services needs a sound security strategy that includes identifying, analyzing, and preventing cybercrime.

Table 1
Essential Tools According to the CIA Triad

Tool	Target
	Confidentiality
Encryption and decryption	Protect data that is sensitive and valuable, such as bank credit card numbers and e-commerce transactions.
Access Control	Establishing access rules for systems, physical elements, and virtual network resources to provide users with access rights and permissions.
Authentication	Granting permissions to verify a user's identity as part of any authentication process.
Authorization	Providing users with control over their actions within a restricted security framework to ensure authorized access to system resources based on predefined policies.
	Integrity
Backups	Storing data in a database management system (DBMS) at regular intervals or automatically.
Checksums	Ensures the integrity of data sent between networks using a mathematical function that converts the contents of a file into a numeric value.
Data correction	Identify unforeseen alterations by retaining genuine data and pinpointing variances between them.
	Availability
Physical protection	Protecting infrastructure elements that transfer their resources stored in a secure area to ensure the safety of accessible data.

Indeed, major cybersecurity techniques such as firewalls, intrusion detection systems (IDS), virtual private networks (VPNs), machine learning for IDS-based intrusion detection, and intrusion prevention systems (IPS) are discussed below.

1. Firewalls and virtual private networks (VPNs) form a strong barrier between network components to provide protection. Typically, a firewall is placed on all sides of a network or subnet to protect it from potential threats from the outside. According to research [7], a network firewall is a collection of elements located between the internal and external networks. It enhances cybersecurity by monitoring all traffic passing through and determining who has permission to access between networks. Firewalls are important elements for securing an internal network, while VPNs are used to manage the rules of access to that network.

It is advisable to utilize a VPN in conjunction with a network firewall to safeguard network traffic. In the absence of a firewall, VPN encryption mechanisms lose their efficacy. Fig. 1 depicts a firewall and a VPN, with the VPN server positioned on the Internet ahead of the firewall.

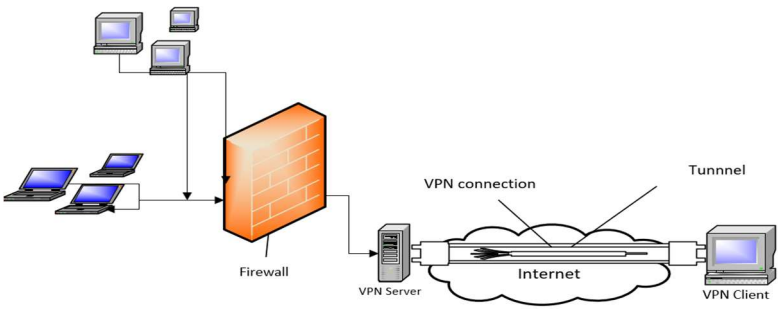

Figure 1: Firewall and VPN architecture on the network

2. An intrusion detection system (IDS) assumes a pivotal role in safeguarding information and data against unauthorized actions that could potentially harm the system. Unauthorized access

to computer systems and networks is deemed illicit activity, posing a direct menace to the confidentiality of information. Therefore, IDS is used to detect the identification of malicious and malicious activities in networks and computer systems, which helps ensure their security.

3. Intrusion detection systems (IDS) based on machine learning techniques represent one of the key problems in the field of cyber research. In today's cybersecurity centers, there is a need to use effective machine learning-based IDS to detect new and sophisticated attacks that can bypass traditional detection methods such as IDS and firewalls. Intrusion Prevention System (IPS) is becoming increasingly important as internetworking grows, which is a critical and dangerous issue. Previously, conventional defenses such as firewalls and other security methods were sufficient to combat traditional cyber threats. In general, IPS functionality is similar to IDS but requires additional precautions. Additionally, compared to IDS, IPS can be divided into two main types: host-level intrusion prevention system (HIPS) and network-level intrusion prevention system (NIPS).

The most common types of cyberattacks were identified and described:

1. Phishing is an illegal practice in which attackers impersonate trusted individuals to commit fraudulent activities. Phishing attacks are aimed at obtaining stolen information, while users, by entering their data, trust that they are dealing with a genuine organization or person.
2. Malicious software (MSW) refers to a category of malicious tools that includes various types of cybercrime such as ransomware, viruses, spyware, and worms. The main goal of malware is to disrupt the network infrastructure identified after the discovery of vulnerabilities in the system.
3. Email spam is a group of unsolicited messages sent by unauthorized companies and organizations. This method allows you to send bulk messages to a large number of users, offering attractive promotions and promotional offers, which may mislead visitors.
4. A distributed denial of service (DDoS) attack is a method of overloading network systems or servers with traffic to exhaust available bandwidth and negatively impact resources. As a result, such attacks temporarily disrupt normally functioning networks or servers.
5. SQL injection represents a type of assault wherein a perpetrator embeds malevolent code into a web server. This attack is typically carried out by injecting a malicious SQL statement through a vulnerable input field in a web application.
6. DNS tunneling is an attack method in which an attacker uses DNS queries and responses within a network. In this method, protocol data is encoded for transmission through DNS, including HTTP and other protocols. DNS tunneling is used to transmit useful information to the DNS server, allowing an attacker to gain access to remote server components.
7. Scripting (XSS) attack is a category of attack that occurs on the client side. In this type of attack, malicious JavaScript code is injected into a website and executed in the user's browser without the user's knowledge.
8. A path traversal attack involves manipulating the path string that determines the location of a directory on the file system. As a result of this attack, the attacker attempts to gain access to protected files or directories located outside the main directory of the website.

4. Intrusion detection system utilizing machine learning techniques

Several intrusion detection systems (IDS) are experiencing weaknesses due to high false positive rates, leading to increasing challenges faced by cyber analysts with negative consequences. This results in threats being missed if they go undetected. The development of IDS has received significant attention from cyber researchers to increase detection rates to minimize false alarms and challenges in identifying new attacks as network patterns change rapidly and new types of attacks emerge continuously. Manual decision-making by human analysts (security analysts) for early detection of intrusions and analysis of large volumes of attacks is considered a challenging task. In addition, traditional intrusion detection systems require the creation of user signatures, which makes them not intelligent enough. It's crucial to automate the detection process without relying on human intervention. To tackle this challenge, scholars have emphasized the development of intrusion detection systems utilizing machine learning methods to achieve dynamic and intelligent detection with superior accuracy. Employing intrusion detection systems with machine learning, as depicted in Fig. 2, offers attack predictions derived from dependable and pre-processed data, trained on a model to deliver precise outcomes [15]. These systems can outperform traditional methods in monitoring malicious activities in real time with a heightened detection rate.

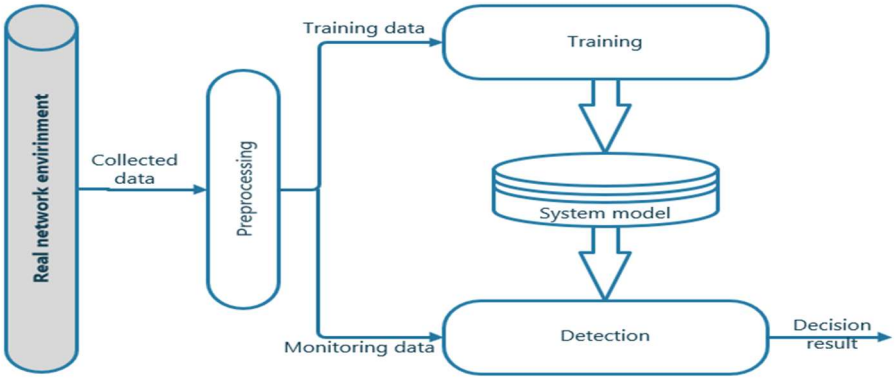

Figure 2: Detection of breaches utilizing machine learning algorithms

A violation detection system based on machine learning typically involves the following essential steps: gathering data through diverse tools like monitoring, logging, and hardware sensors, processing data and features (e.g., regularization, data cleansing, and normalization), modeling phase (e.g., classification, clustering), and assessing performance metrics.

5. HIDS Architecture Analysis

This section examines the deployment process of a host-based intrusion detection system. The data was carefully selected to align with the log file obtained from the web server. Furthermore, a preprocessing approach was implemented alongside the chosen feature representation technique.

Every unprocessed text segment is transformed into a feature vector by encompassing all distinct words found in a lexicon. Consequently, each word in the lexicon serves as a feature vector to depict the input text. An intrusion detection system was explored to execute an attack detection procedure utilizing various machine learning approaches, depicted in Fig. 3.

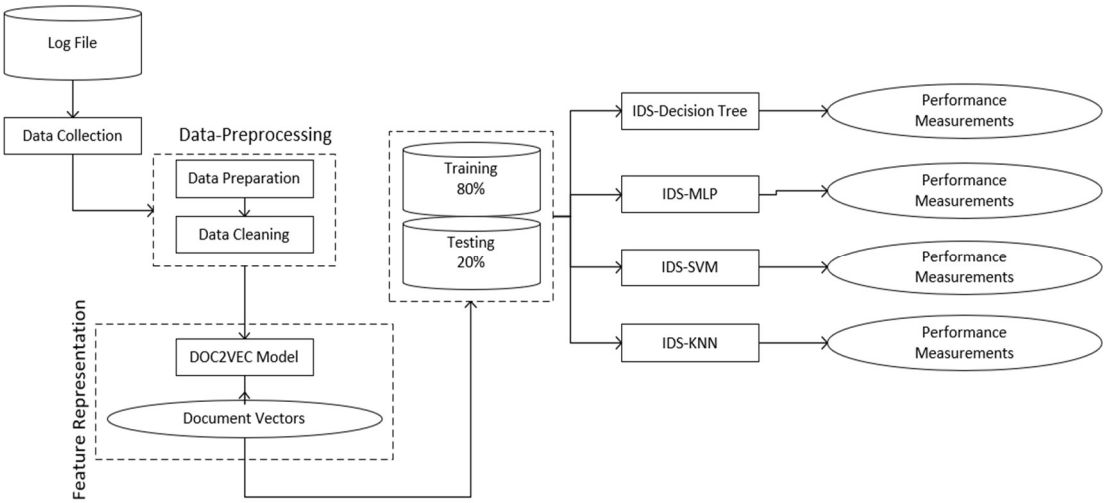

Figure 3: Proposed HIDS architecture with implementation phase

Data collection. Almost all real web servers keep logs, which provide us with valuable material for analysis. In our study, we used the Apache log file, which is in CLF format.

Pre-processing of data. User activity on a website is recorded as strings in web server logs, and some of these entries do not contain useful information. They merely introduce interference and lack relevance for analysis, potentially impairing the performance of attack detection. Hence, before employing machine learning algorithms on the data, a preprocessing phase is imperative.

Data preparation. As the log file comprises unorganized text, it's necessary to convert the log entries into a structured format. This structured representation allows for the extraction of user navigation patterns, facilitating the training of a learning model [16]. Earlier log analysis techniques overlooked the timestamps associated with log entries and relied on a fixed log key to identify abnormal values.

Data cleaning. During the data cleansing process, we change the storage location of the generated CSV file from the previous step to ensure that we only collect the necessary information. In this step, the initial action involves decoding HTTP GET requests into ASCII characters, converting them to lowercase, and eliminating numeric values. Furthermore, a novel method has been suggested for screening HTTP GET requests, comprising the subsequent procedures:

- Returning a raw status code containing the number 200, without a query string.
- Filtering static queries such as ('jpg', 'png', 'gif', 'webp', 'cr2', 'tif', and others).

6. Classification Preliminaries

We sought the optimal machine learning techniques suitable for our dataset to deploy an Intrusion Detection System (IDS) aimed at recognizing prevalent attacks targeting web servers.

In this approach, we applied a document embedding method using the doc2Vec model to extract 4000 document vectors trained by this model. Additionally, we used four different algorithms (Decision Tree, Support Vector Machine, Multilayer Perceptron (MLP), and KNeighbors classifier) to classify the extracted URLs present in HTTP GET requests to detect potentially malicious attacks.

To assess the efficiency of the utilized models, we partitioned the dataset into two segments: 80% for training purposes and 20% for testing purposes. Subsequently, the intelligent models underwent testing using diverse evaluation metrics including accuracy, specificity, area under the ROC curve, and confusion matrix. These metrics were computed for each machine learning classifier, offering vital insights into both existing and detected attack classes.

Additionally, the process of training for each model uses several performance parameters for the machine learning classifier, described below:

1. Decision tree model. A decision tree is a structure similar to a tree, consisting of nodes that form a rooted tree. The main goal of the ID3 algorithm is to construct decision trees using a flowchart, starting from top to bottom. It analyzes every parameter at every node, such as entropy for exploratory analysis. This property distinguishes the learning stage based on their target classification. To achieve this, entropy is checked using the formula (1).

$$Entropy = -\sum_{i=1}^{n} p_i \cdot \log(p_i), \quad (1)$$

where i is the number of all classes and pi is the probability of class I.

- Criterion=Entropy
- Splitter=Best
- Max_depth=None
- Class_weight=Balanced
- Max_features=No.

2. The multilayer perceptron (MLP) model represents one of the latest advancements within the realm of artificial intelligence. Artificial neural networks (ANNs) stand at the forefront of technological innovation in the field of AI. These networks mimic the human brain's capacity to process information, leveraging a vast network of interconnected neurons that operate concurrently. This capability enables ANNs to effectively tackle diverse scientific challenges and derive data-driven insights. Within a neural network, a multilayer perceptron comprises multiple layers, each fulfilling distinct functions. These layers encompass an input layer, hidden layers, and an output layer, also referred to as the source layer. The mathematical MLP classifier as shown in Fig. 4 [17] is calculated using the Rectified activation function Linear Unit (ReLU) (2).

$$ReLU(x) = \max(x, 0), \quad (2)$$

where x represents an element, and ReLU (x) denotes the function that returns the maximum value of that element or zero.

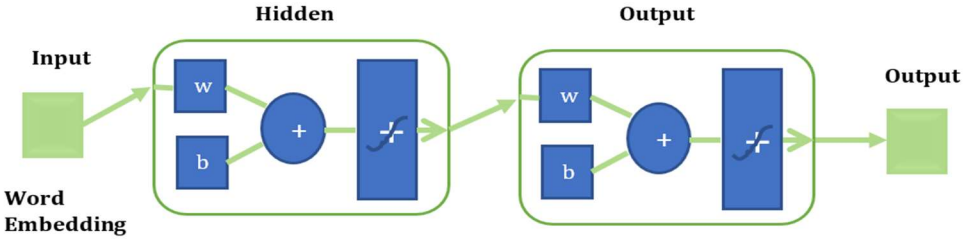

Figure 4: Artificial neural network: The Multi-Layer Perceptron (MLP) model

Consequently, our model was evaluated using word vectors, which served as the input layer, hidden layer, and output layer, with the parameters outlined below:

- Activation: Relu is employed as the rectified linear unit function.
- Solver: Adam is used as the solver for weight optimization.
- Hidden_layer_sizes: Set to 6, representing the number of neurons in the hidden layer.
- Learning rate: Fixed at 0.001 to schedule weight updates.
- Random_state: Employing batch sampling to precisely initialize random numbers for weights and biases.
- Momentum: Set at 0.9 as the gradient descent update.
- Alpha: Maintained at 0.0001 to prevent overfitting by penalizing weights.
- Max_iter: Specified as 1000, denoting the maximum number of iterations; the solver continues iterating until convergence is reached within this set number of iterations.

3. Support Vector Machine (SVM): This learning algorithm functions by initially separating classes that are linearly separable using hyperplanes, and then extending this concept to nonlinear boundaries by altering the space. In this model, hyperplanes (decision limits) are calculated using the formula $h(x)=x^T\beta+\beta_o=0$. Then the distance from the point to the hyperplane is calculated using the formula $d(x)=\frac{x^T\beta+\beta o}{||\beta||}$, where $||\beta||=\sqrt{\beta_1^2+\cdots+\beta_p^2}$ and maximizing the width of the dividing strip is equivalent to minimizing the norm of the parameter vector β.

Support vector machine (SVM) can be applied in two manners: for addressing classification and regression issues. In our model, the subsequent parameters were utilized during both the training and testing phases:

- Kernel=linear: with the equation: $k(x\,i, x\,j)=x\,i.\,x\,j$
- Tol=0.001 Tolerance level for the stopping criterion: 1e-3
- gamma='scale' that uses 1/(n _features * X.var ( )) as the value of gamma
- cache_size=200 as dimension of the kernel cache
- max_iter=-1 for no limit.

4. K-Nearest Neighbors (KNN) Method: In this model, we used supervised learning using the KNN technique, which is used for classification problems. This method operates on the hypothesis that data points of the same class are close to each other.

To identify attack classes, the following steps were performed:

- Step 1: The K value must be determined.
- Step 2: We calculated the distances between data points in the test and training sets using the Minkowski metric.
- Step 3: Distances were arranged to determine the k nearest neighbors, prioritizing those with the smallest distance values.
- Step 4: We examined the neighbors to ascertain the suitable category for new data (test) by considering the majority of votes.
- Stage 5: Final classification achieved.

In this approach, we evaluated the efficacy of the trained model utilizing the subsequent parameters:

- Weight: uniform function applied in the detection process.
- Metric: Minkowski with parameter p=2, which corresponds to the standard Euclidean metric.
- n_neighbors: 2—the number of neighbors used for calculations.
- Algorithm: 'auto'—algorithm used to determine nearest neighbors.
- n_jobs: 1—the number of parallel tasks, allowing for searching for neighbors.

In the following section, we will delve into the outcomes of each model to conduct a comparative assessment. This will enable us to identify the most precise and effective model among the chosen machine learning methodologies.

7. Experimental results and their analysis

The effectiveness of the examined Host-based Intrusion Detection System (HIDS) was assessed using four machine learning methods: decision trees, multilayer perceptron (MLP), support vector machines, and the K-nearest neighbor classifier. Our objective is to recognize the most harmful attacks a web server might face.

To evaluate the applied HIDS, we analyzed various evaluation metrics including detection rate, precision, recall, F-measure, area under the ROC curve, and error matrix. It is noted that when using each of the considered algorithms, the results of each model differ significantly.

Therefore, Table 2 summarizes the detection rates for each attack tested using the proposed IDS.

Table 2
Maximum detection accuracy achieved by classifiers for each attack

Classifier	Attack type		
	Path-Traversal	SQLI	XSS
SVM	0.9418	0.8653	0.6224
MLP	0.9137	0.9346	0.8277
Decision Tree	0.8091	0.9092	0.8197
KNN	0.9061	0.9207	0.7717

We can conclude that the SVM classifier achieves the highest accuracy in detecting the Path-Traversal attack, being 94.18%. The MLP classifier achieves maximum detection rates for SQLI and XSS attacks of 93.46% and 82.77%, respectively. On the other hand, SVM shows the best detection rate for path traversal attacks, although it shows the worst result in XSS detection at 62.24%.

We also calculated the area under the ROC curve for each model using true positives and false positives. A high value of the area under ROC indicates a high ability of the model to detect intruders, while a low value indicates that the model is insufficiently effective in this regard.

Regarding the area under the ROC curve of each model, we can note that MLP classifiers achieved the highest score of 93%, while KNN and Decision Tree achieved 90% and SVM achieved the lowest score of 87%.

By examining 4000 records contained in our dataset, the average accuracy score of each selected classifier was calculated and presented in Table 3. We can conclude that the MLP classifier shows the highest accuracy with a result of 89.44%, while the SVM shows the lowest accuracy of 81.44%.

After conducting the analysis using the decision tree model and examining the data from Table 3 in Fig. 5, it was found that the decision tree model performed average on the test data.

Table 3
Classification report

Machine Learning Classifiers	Accuracy	Precision/Recall	Path Traversal	SQLi	XSS	Macro avg	Weighted avg
Decision Tree model	0.8485	Precision	0.8920	0.8797	0.9007	0.7638	0.8481
		Recall	0.8214	0.8091	0.9092	0.8197	0.8460
MLP model	0.8944	Precision	0.8747	0.8624	0.9302	0.8877	0.8934
		Recall	0.9260	0.9137	0.9373	0.8277	0.8920
SVM model	0.8144	Precision	0.7571	0.7448	0.8776	0.8407	0.8211
		Recall	0.9541	0.9418	0.8653	0.6224	0.8098
KNN model	0.8694	Precision	0.8511	0.8388	0.9229	0.8399	0.8672
		Recall	0.9184	0.9061	0.9207	0.7717	0.8662

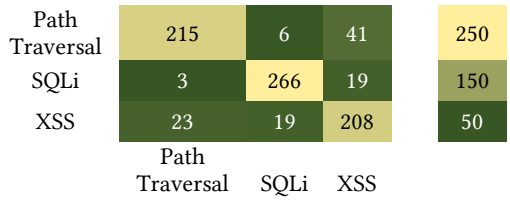

Figure 5: Confusion matrix for the decision tree model

The path traversal attack showed an accuracy of 87.97% and a recall rate of 80.91%, while SQLI showed the highest accuracy rate of 90.07% and a recall rate of 90.92%. XSS, in turn, was the least accurate with a precision of 76.38% but a recall rate of 81.97%. The decision tree model achieved an average accuracy of 84.81% and the area under the ROC curve achieved an average of 90%.

Regarding the MLP model, examination of Table 3 and Fig. 6 leads to the conclusion that this model achieved improvement during the testing process.

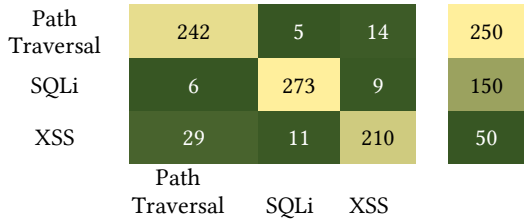

Figure 6: Confusion matrix for the MLP model

Thus, the path traversal attack showed low accuracy (86.24%), SQLI—high (93.02%), and XSS—88.77%. The recall rates for path, SQLI, and XSS attacks were 91.37%, 93.73%, and 82.77%, respectively. Overall, this model increased the average detection accuracy to 89.34%, and the area under ROC reached an average value of 92%.

Additionally, it was noted that the SVM model showed expansion in some classes during testing after analyzing Table 3 and Fig. 7.

Figure 7: Error matrix for the SVM model

For example, the path traversal attack achieved a low accuracy rate of 74.48%, while SQLI and XSS achieved accuracy rates of approximately 87.76% and 84.07%, respectively. Feedback for Path, SQLI, and XSS attacks were rated 94.18%, 86.53%, and 62.24%. Additionally, the average accuracy was 83.11% and the average area under the ROC curve was 87%.

Finally, after analyzing the data from Table 3 and Fig. 8, it can be concluded that the KNN model showed acceptable results during testing, with the path traversal attack achieving a low accuracy of 83.88%.

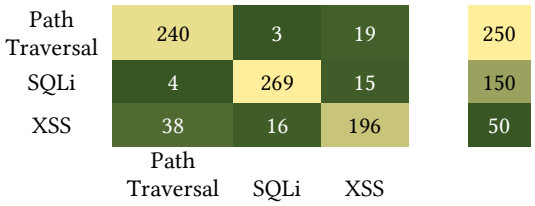

Figure 8: Confusion matrix for the KNN model

The SQLI attack achieved high accuracy at 92.29%, while the XSS attack achieved 83.99%. The responses to path following, SQLI, and XSS attacks were 90.61%, 92.07%, and 77.17%, respectively. The average accuracy value in the detection phase for this model was 86.72%, and the area under the ROC curve had an average value of 90%.

8. Conclusions

In this research work, a mechanism was proposed to enhance security in the university teaching laboratory system. To achieve this, a penetration testing methodology was applied by the OWASP Top 10, which aims to identify vulnerabilities associated with common attacks such as SQL injection, cross-site scripting, and leakage of sensitive data. Security solutions and recommendations were presented that can be used as a guide by IT administrators to protect the system from cybercrime threats. Thus, the effectiveness of the proposed system was proven by eliminating the detected vulnerabilities to achieve basic security standards. A host-based intrusion detection system (HIDS) using text mining techniques has also been developed.

A dataset was compiled containing 4000 entries of malicious URLs associated with HTTP GET requests. A complex hypothesis is formulated involving feature representation methods with three main limitations: the difficulty of obtaining URL information, the need for manual work to extract important features, and the inability to obtain useful information about hidden HTTP GET requests. To overcome these limitations, the Doc2Vec model was proposed as a feature representation method in our Host-Based Intrusion Detection System (HIDS). In addition,

four different machine learning methods were used (KNN, SVM, Decision Tree, and MLP) to propose the best and most efficient classification model. Our HIDS system has demonstrated the ability to detect SQLi, XSS, and directory traversal attacks. Thus, MLP showed the best accuracy of 89.44%, followed by KNN with 86.94% accuracy, and then decision tree with 84.85%. Finally, SVM showed the lowest level of accuracy at 81.44%.

References

[1] V. Dudykevych, et al., Platform for the Security of Cyber-Physical Systems and the IoT in the Intellectualization of Society, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, CPITS, vol. 3654 (2024) 449–457.

[2] O. Shevchenko, et al., Methods of the Objects Identification and Recognition Research in the Networks with the IoT Concept Support, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 2923 (2021) 277–282.

[3] P. Anakhov, et al., Protecting Objects of Critical Information Infrastructure from Wartime Cyber Attacks by Decentralizing the Telecommunications Network, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3550 (2023) 240–245.

[4] S. Popereshnyak, A. Vecherkovskaya, V. Zhebka, Intrusion Detection based on an Intelligent Security System using Machine Learning Methods, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3654 (2024) 163–178.

[5] A. K. Kassem, Intelligent system using machine learning techniques for security assessment and cyber intrusion detection (2021).

[6] J. Fonseca, M. Vieira, H. Madeira, Evaluation of Web Security Mechanisms using Vulnerability & Attack Injection, IEEE Transactions on Dependable & Secure Computing 11 (2013) 440–453.

[7] Imperva, Web Application Attack Report WAAR (2015). URL: https://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed6.pdf

[8] I. Hydara, et al., Current State of Research on Cross-site Scripting (XSS)—A Systematic Literature Review, Information and Software Technology, 58 (2015) 170–186.

[9] Y. Donga, Y. Zhanga. Adaptively Detecting Malicious Queries in Web Attacks, ArXiv, 1701.0777 (2017).

[10] A. K. Kassem, et al., A Survey of Methods for the Construction of an Intrusion Detection System, Artificial Intelligence and Applied Mathematics in Engineering Problems, 43. (2019) 211–225.

[11] V. Zhebka, et al., Optimization of Machine Learning Method to Improve the Management Efficiency of Heterogeneous Telecommunication Network, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3288 (2022) 149–155.

[12] V. Zhebka, et al., Methodology for Predicting Failures in a Smart Home based on Machine Learning Methods, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, CPITS, vol. 3654 (2024) 322–332.

[13] V. Buhas, et al., Using Machine Learning Techniques to Increase the Effectiveness of Cybersecurity, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3188, no. 2 (2021) 273–281.

[14] Deep Software "Deep Log Analyzer" (2018). URL: https://www.deep-software.com/

[15] S. Qadir, S. M. K. Quadri, Information Availability: An Insight into the Most Important Attribute of Information Security, J. Inf. Secur. 7 (2016) 185–194.

[16] J. G. Lou, et al., Mining Invariants from Console Logs for System Problem Detection, USENIX Annual Technical Conference (2010) 24–24.

[17] M. J. H. Mughal, Data Mining: Web Data Mining Techniques, Tools and Algorithms: An Overview, Int. J. Adv. Comput. Sci. Appl. 9(6) (2018). doi: 10.14569/IJACSA.2018.090630.