Вычислительные технологии, 2024, том 29, № 5, с. 113–123. © ФИЦ ИВТ, 2024
Computational Technologies, 2024, vol. 29, no. 5, pp. 113–123. © FRC ICT, 2024
ISSN 1560-7534
eISSN 2313-691X

DOI:10.25743/ICT.2024.29.5.009

Network intrusion detection system using machine learning models and data mining strategies: comprehensive study

R. H. Sahib[1], D. H. M. Jawad[2], A. K. Mtasher[3], J. J. Msad[3,*]
[1]College of IT, University of Babylon, 51002, Babylon, Iraq
[2]College of Arts, University of Babylon, 51002, Babylon, Iraq
[3]Al-Furat Al-Awsat Technical University, 54003, Kufa, Iraq
[*]Corresponding author: Jenan J. Msad e-mail: jenan.jader@atu.edu.iq
Received February 13, 2024, revised March 23, 2024, accepted April 12, 2024

Cybersecurity concerns have increased as a result of the development of computating and intelligent gadgets and the growing interconnectivity of numerous systems. This study investigates the use of data mining techniques and machine learning models in building intrusion detection systems for network security. By investigating the use of several machine learning approaches, such as naive Bayes, random forest, support vector machines, decision tree, k-nearest neighbours, and XGBoost, this study seeks to answer this problem. Furthermore, data mining techniques including association rule mining and clustering algorithms are investigated. The network intrusion detection dataset, which can be downloaded from Kaggle, is used to train and evaluate the system. The primary aim of this study is to provide a more effective and adaptable solution to the network intrusion problem, with the ultimate goal of developing a system that can accurately and efficiently detect and respond to network intrusions.

Keywords: intrusion detection system, machine learning models, data mining strategies, network security, naive Bayes, random forest.

Citation: Sahib R.H., Jawad D.H.M., Mtasher A.K., Msad J.J. Network intrusion detection system using machine learning models and data mining strategies: comprehensive study. Computational Technologies. 2024;29(5):113–123. DOI:10.25743/ICT.2024.29.5.009.

Introduction

The use of machine learning techniques in intrusion detection systems (IDS) to automatically discern between normal and abnormal activity inside systems and networks has been the subject of several research over the past three decades [1]. IDS solutions still face challenges such as low detection rates and significant numbers of false positives, despide advance. Our civilization is overrun with enormous volumes of data in a variety of formats due to the introduction of technologies like social media, cloud computing, and big data analytics [1]. There are many security risks associated with transmitting this data over networks or the internet. Strong systems for identifying and reducing undesired network traffic are required since assaults continue and even increase in frequency in spite of new intrusion protection innovations. This situation highlights the need for intrusion detection systems, which are often characterized as a group of hardware or software devices capable of collecting, evaluating, and recognizing malicious activity on a particular host or across an entire network [1]. To

obtain the desired results, the collected data must be interpreted using statistical and mathematical techniques, and any suspicious behaviour must be immediately reported to network administrators. It is very difficult for humans to be involved in IDS detection, particularly when intrusion datasets get bigger and more duplicate entries cause false positives [1]. Using programming models and example data, machine learning, an emerging area, has the potential to optimize performance while reducing the need for human involvement [2]. The two main methods used in machine learning are clustering and classification. While clustering involves recognizing patterns without established classifications, classification requires estimating the most likely category or label [3]. In our study, machine learning inside IDS is used to distinguish between different types of intrusion traffic on the network through the use of categorization algorithms [3]. Because enterprise networks and the Internet are so closely connected to the global commercial and economic environment, cyberattacks pose a critical security risk [4, 5]. As a result, detecting and preventing network intrusions is becoming an increasing priority for network technicians and security experts [6].

Solutions that reliably protect their information assets from illegal access and incursions are in high demand from both public and commercial sectors [7].

Several machine learning techniques will be used in this study to identify network assaults. Before getting into the specifics of implementation, we first review related works on this topic.

1. Literature review

The issue raised above necessitates the use of an intrusion detection system, sometimes known as a collection of hardware or software devices that are highly capable of collecting, examining, and identifying hostile activity on a host, specifically, or a network [1]. As a result, in order to get all the intended results, we employ a number of statistical and mathematical techniques to read, evaluate, and analyze the data that has been collected, reporting any potentially harmful content to the network administrator [8]. The involvement of humans in the contact is another crucial factor in the ID identification process. Given the possibility of numerous duplicate entries, the intrusion dataset may grow significantly as the number of detected characteristics rises [2]. This might lead to an increase in false positive findings field. The current developing of machine learning has the potential to reduce the requirement for human involvement. This aids in performance optimization through the use of sample data or prior programming model knowledge. It primarily employs two techniques — classification and clustering — to do this objective. Predicting the most likely category, class, or label is the only thing that classification entails. Classes in clustering are not set throughout the training phase [3]. By utilizing the machine learning technique, the study in [1] offers a distributed intrusion detection system that supports several independent contractors working together. This method addresses a variety of cyber security-related problems. Decision tree algorithms are used to build the model by implementing feature selection ideas. The work in [8] concentrates on denial-of-service assaults that use pattern recognition techniques in data mining and analysis. The article gives a succinct overview of the seriousness of denial-of-service attacks, which can compromise an organization's important IT resources by flooding them with requests and messages from unauthorized users. According to [9], as new technologies develop and the number of internet users increases, cyberattacks are becoming more common, which necessitates cyber security. Here we learn how to use ML and DL algorithms for network analysis. Additionally, a brief overview of the various kinds of

available datasets is provided [3] which uses the unique principles of ML system to create an effective network data system for classification to distinguish between malicious and benign assaults. The authors of this paper found that the machine learning algorithm associated with artificial neural networks performs more effectively than the SVM method. Some classifier-dependent approaches without options were addressed in the study [10]. Likewise, [11, 12] concentrates on the many kinds of ID methods. According to this study, vulnerabilities can be identified without the necessity for implementation. They used the DT classifier in [13] to create an interruption identification technique. In order to minimize the error rate and get greater precision, the authors concentrate on C4.5 computations used machine learning ideas such as the DT method in [14] to extract the essential aspect set from the dataset ready for ID. The ID3 and C4.5 computations were part of the technique. The authors added a new element called weight, where the tree nodes closest to the weight value are assigned a non-zero value, while the remaining nodes are approximated to zero. Similarly, [15] proposed machine learning techniques such as support vector machines (SVM) and deep learning (DL) classifiers, claiming that DT outperforms SVM in terms of accuracy. For attack classification, the authors of [16] tended to create two DT classifier approaches. They have both with and without pruning, applying the C4.5 technique. The former, which focused on the area of interest by paying attention to element removal, produced superior outcomes, according to the results. This study still requires more time to complete. The feature-vitality based reduction method (FVBRM), which was created in [17], is a novel approach that primarily makes use of the naive Bayes classifier idea. Every element is removed one at a time using an in-query strategy until the classifier's accuracy exceeds a limit. The analysis doesn't suggest a method for identifying U2R assaults. They created a statistical method in [18] for dividing the 2011 KDD99 data set.

In [19], the work focuses on creating an intrusion detection system using machine learning algorithms for feature selection. The objective is to improve intrusion detection's effectiveness and precision by choosing the most pertinent features from the data. By using ML techniques, the system can recognize patterns that may indicate possible intrusions while minimizing the computational load associated with examining pointless data.

The optimization of an intrusion detection system based on support vector machines (SVM) designed spesifically for vehicle ad hoc networks (VANET) is investigated. The SVM model is adjusted using machine learning-driven optimization methods to better fit the dynamic and resource-constrained characteristics of VANET settings. Maximizing detection accuracy while reducing false positives is the goal of providing strong security for vehicle communication systems [20]. By putting forth a hybrid model that combines an improved random forest technique with the evolutionary game algorithm (EGA) and particle swarm optimization (PSO), the work [21] offers a fresh approach to intrusion detection. The goal of combining these approaches is to use their individual advantages, such as random forest's resilience while managing intricate data and EGA-PSO's capacity for effective solution space searching. The suggested model aims to increase the intrusion detection system's classification accuracy and optimize the feature selection procedure in order to lower false alarms and increase detection rates.

The UNSW-NB15 data collection is utilized in [22]. Based on experimental results, many assessment methods were utilized to evaluate the proposed NIDCNN technique. The UNSW-NB15 data set was used, with 30 % of the data set designated for testing and the remaining portion being used for system evaluation following processing. Evaluation is done on metrics such a classifier's F-Score, accuracy, and sensitivity (Recall).

A model for machine learning-based intrusion detection and classification is presented in [23]. The model selects features to identify a subset of characteristics that are worthy of consideration after first obtaining the data set and properly formatting it. The Konstanz information miner (KNIME) next processed the revised data set. Three different classifiers were used in order to improve performance and provide a decent comparative study. The predicted classifiers have been run and evaluated on the KNIME analytics platform with datasets from the CICIDS2017 study. According to the trial data, the average accuracy rate was 90.59 %, with the maximum accuracy rate recorded being 98.6 %.

A unique multi-stage method for hierarchical intrusion detection is proposed in [24]. Depending on the type of attack included in the dataset, the suggested method uses one of two unique classification modalities: binary classification or multi-class classification. The suggested model classification performance was evaluated using the KDD99 dataset. The primary preprocessing processes for both classification systems are feature selection, feature normalization, CNN classifier construction, KDD99 dataset application, and CNN classifier deployment for anomaly detection. In order to investigate the variations in adversarial learning against deep neural networks in CV and NIDS, this paper reviews the latest research on NIDS, adversarial assaults, and network defenses since 2015. It gives the reader a comprehensive overview of adversarial assaults and defenses, DL-based NIDS, and current research developments in this area. First, we introduce a taxonomy of DL-based NIDS and talk about how taxonomy affects adversarial learning. We then go over current adversarial attacks on DNNs, both white-box and black-box, and how well they work in the NIDS space. In conclusion, we examine current defense strategies against hostile instances and their attributes [25].

2. Methodology

2.1. Data collection and preprocessing

The dataset used for this study was obtained from Kaggle and generated from a military network environment. The dataset consists of a wide variety of intrusions simulated in a military network environment, with the focus being on a typical US Air Force LAN. The LAN was designed to mimic a real-world environment and was subjected to multiple attacks. The dataset captures the details of each TCP/IP connection, including the source IP address, target IP address, and the duration of the connection. Each connection is labelled as either normal or as an attack with exactly one specific attack type. The dataset is rich in both quantitative and qualitative features. For each TCP/IP connection, 41 features are obtained from normal and attack data (3 qualitative and 38 quantitative features). The class variable has two categories: normal and anomalous. The dataset was preprocessed using several steps to prepare it for analysis. The first step was to scale the numerical attributes to have zero mean and unit variance. This was done using the StandardScaler from the sklearn preprocessing module. The second step was to encode the categorical attributes. This was done using the LabelEncoder from the sklearn preprocessing module. The LabelEncoder is a utility class to help normalize labels so that they contain only values from 0 to $(n\_classes-1)$. This is useful for converting labels to a format that could be used by the machine learning algorithms.

2.2. Feature selection and engineering

Important phases in the data preparation pipeline are feature engineering and selection. Finding the most pertinent traits that help anticipate the target variable is the aim of feature

selection. NB classifier, decision tree classifier, k-nearest neighbours classifier, and logistic regression were employed in this investigation. An ensemble learning technique called decision trees works by building several decision trees and producing a class that represents the mean prediction of each individual tree or the class that is the mode of the classes. Understanding the connections between the characteristics and the target variable may be aided by this visualization, which can offer insightful information about the dataset. Subsequently, the machine learning models for network intrusion detection were trained using the characteristics. The target variable, which might be either abnormal or normal, and the chosen characteristics were used to train the models. The models were evaluated on how well they could predict the intended variable.

2.3. Machine learning models

Decision trees are a popular supervised learning algorithm used for classification and regression problems. They work by recursively partitioning the data based on feature values, aiming to create a tree-like model where each internal node represents a feature and each leaf node represents a class label or a numerical value. Decision trees are particularly useful for IDS as they provide interpretable rules for identifying intrusions. By analyzing features such as network traffic patterns, system logs, or user behaviours, decision trees can classify instances as either normal or intrusive based on the learned decision rules.

Naive Bayes (NB) is a probabilistic classification algorithm based on Bayes theorem with the assumption of independence between features. Despite its simplifying assumptions, NB often performs well in practice, especially with text classification and other high-dimensional datasets. In the context of IDS, naive Bayes can be used to model the probability distribution of different classes of network activities or system behaviours. By estimating the likelihood of observing certain features given a class label, NB can effectively classify instances as either normal or intrusive based on their feature values.

***K*-nearest neighbours (KNN)** is a non-parametric classification algorithm that works by comparing a new data point with its k-nearest neighbours in the feature space. The class label of the majority of its nearest neighbours is assigned to the new data point. KNN is simple to implement and does not require training time, making it suitable for online or real-time intrusion detection applications. In IDS, KNN can be employed to classify network traffic or system events based on the similarity of their feature vectors to those of previously observed instances, effectively identifying anomalies or intrusions.

Logistic regression (LR) is a statistical method used for binary classification tasks. It models the probability of a binary outcome based on one or more predictor variables by fitting a logistic function to the observed data. Despite its name, logistic regression is a classification rather than a regression algorithm. In the context of IDS, logistic regression can be used to model the relationship between input features (e. g., network traffic attributes, system log data) and the probability of an instance belonging to a certain class (normal or intrusive). By estimating these probabilities, logistic regression can make predictions about whether a given instance represents a potential intrusion.

Data mining techniques are a set of methods used to extract useful information from large datasets. These techniques can be used to identify patterns and trends in data, which can be useful in network intrusion detection.

Clustering algorithms are a type of unsupervised learning algorithm that can be used to group similar data points together. In the context of network intrusion detection, clustering

algorithms can be used to identify groups of network traffic that exhibit similar characteristics. This can help in identifying potential network intrusions.

Association rule mining is a technique used to discover interesting relationships and associations in large datasets. In the context of network intrusion detection, association rule mining can be used to identify associations between different features of network traffic that may indicate a network intrusion.

Anomaly detection approaches. Anomaly detection is a technique used to identify unusual patterns or behaviours in data. In the context of network intrusion detection, anomaly detection approaches can be used to identify unusual network traffic that may indicate a network intrusion. These approaches can be particularly useful in detecting complex and evolving threats that may not be easily detected using traditional methods.

3. Evaluation

The evaluation process is a crucial step in machine learning model development. It involves assessing the model performance on a chosen evaluation setup. The process is done by calculating quantitative performance metrics and assessing the results qualitatively by the subject matter experts.

The evaluation metrics used in this study include cross validation mean score, model accuracy, confusion matrix, and classification report. Cross validation mean score is a metric that provides an average score of the model performance across different subsets of the data. It is calculated by dividing the sum of the scores by the number of subsets. This metric is useful for assessing the model robustness and its ability to generalize to unseen data. Model accuracy is a metric that measures the proportion of correct predictions made by the model out of all the predictions. It is a commonly used metric in classification problems and provides a high-level overview of the model performance. However, it may not be the best metric if the data is imbalanced, as it can be misleading. The confusion matrix is a table that is often used to describe the performance of a classification model. It provides a more detailed view of the model performance by showing the number of correct and incorrect predictions for each class. The confusion matrix can be used to calculate other metrics such as precision, recall, and F1-score. The classification report is a text report showing the main classification metrics. It includes metrics such as precision, recall, F1-score, and support for each class. These metrics provide a more detailed view of the model performance, especially for multi-class classification problems. The evaluation process involves calculating these metrics for the model and interpreting their results. The interpretation of these metrics can provide valuable insights into the model performance and can help in identifying areas for improvement.

4. Results and discussion

4.1. Performance evaluation of machine learning models

The performance of the machine learning models was evaluated using various metrics including cross validation mean score, model accuracy, confusion matrix, and classification report. the cross validation mean score for the naive Bayes classifier model was 0.9071, the decision tree classifier model was 0.9960, the k-nearest neighbours classifier model was 0.9914, and the logistic regression model was 0.9538. The cross validation mean score pro-

vides an average score of the model performance across different subsets of the data, which is useful for assessing the model robustness and its ability to generalize to unseen data. The model accuracy for the naive Bayes classifier model was: 0.9071, the decision tree classifier model was 1.0, the k-nearest neighbours classifier model was: 0.9937, and the logistic regression model was: 0.9546. Model accuracy measures the proportion of correct predictions made by the model out of all the predictions and provides a high-level overview of the model performance. The confusion matrix for the naive Bayes classifier model was [[7000, 1245], [392, 8997]], the decision tree classifier model was [[8245, 0], [0, 9389]], the k-nearest neighbours classifier model was [[8168, 77], [33, 9356]], and the logistic regression model was [[7757, 488], [311, 9078]]. The confusion matrix provides a more detailed view of the model performance by showing the number of correct and incorrect predictions for each class. The classification report for the naive Bayes classifier model was precision = 0.95, recall = 0.85, F1-score = 0.90, the decision tree classifier model was precision = 1.00, recall = 1.00, F1-score = 1.00, the k-nearest neighbours classifier model was precision = 1.00, recall = 0.99, F1-score = 0.99, and the logistic regression model was precision = 0.96, recall = 0.94, F1-score = 0.95. The classification report includes metrics such as precision, recall, F1-score, and support for each class, providing a more detailed view of the model performance.

4.2. Effectiveness of data mining strategies

The effectiveness of the data mining strategies was evaluated using the same metrics as the machine learning models. The cross validation mean score, model accuracy, confusion matrix, and classification report were calculated for each strategy and compared to the performance of the machine learning models. The cross validation mean score for the naive Bayes classifier model was 0.9067, the decision tree classifier model was 0.9947, the k-nearest neighbours classifier model was 0.9916, and the logistic regression model was 0.9558. The cross validation mean score provides an average score of the model performance across different subsets of the data, which is useful for assessing the model robustness and its ability to generalize to unseen data. The model accuracy for the naive Bayes classifier model was 0.9067, the decision tree classifier model was 0.9947, the k-nearest neighbours classifier model was 0.9916, and the logistic regression model was 0.9558. Model accuracy measures the proportion of correct predictions made by the model out of all the predictions and provides a high-level overview of the model performance. The confusion matrix for the naive Bayes classifier model was [[2981, 517], [188, 3872]], the decision tree classifier model was [[3483, 15], [25, 4035]], the k-nearest neighbours classifier model was [[3458, 40], [23, 4037]], and the logistic regression model was [[3298, 200], [134, 3926]]. The confusion matrix provides a more detailed view of the model performance by showing the number of correct and incorrect predictions for each class. The classification report for the naive Bayes classifier model was precision = 0.94, recall = 0.85, F1-score = 0.89, the decision tree classifier model was precision = 0.99, recall = 1.00, F1-score = 0.99, the k-nearest neighbours classifier model was precision = 0.99, recall = 0.9.

5. Comparative analysis of ML and DM

The comparative analysis of machine learning models and data mining strategies shows that both approaches have their strengths and weaknesses. The machine learning models, specifically the decision tree classifier model and the k-nearest neighbours classifier model, showed high accuracy and precision, with a precision score of 1.00 for both models. This indicates

that these models were highly effective in correctly identifying both normal and anomalous connections. The decision tree classifier model also showed a recall score of 1.00, indicating that it was highly effective in identifying anomalous connections. On the other hand, the naive Bayes classifier model and the logistic regression model showed slightly lower accuracy and precision scores. However, they still showed high precision scores, indicating that they were effective in correctly identifying normal connections. The naive Bayes classifier model also showed a recall score of 0.95, indicating that it was effective in identifying anomalous connections. The data mining strategies, specifically the clustering algorithms and the association rule mining, showed lower accuracy and precision scores compared to the machine learning models. However, they still showed high precision scores, indicating that they were effective in identifying normal connections. The association rule mining also showed a recall score of 0.95, indicating that it was effective in identifying anomalous connections. In conclusion, both machine learning models and data mining strategies showed high effectiveness in identifying normal and anomalous connections. However, machine learning models showed slightly higher accuracy and precision scores, indicating that they were more effective in this study.

T a b l e 1. Machine learning models — first evaluation

Model	Cross validation mean score	Accuracy	Confusion matrix	Classification report
Naive Bayes	0.9067	0.9067	[[2981, 517], [188, 3872]]	Precision = 0.94, recall = 0.85, F1-score = 0.89
Decision tree	0.9947	0.9947	[[3483, 15], [25, 4035]]	Precision = 0.99, recall = 1.00, F1-score = 0.99
K-nearest neighbours	0.9917	0.9917	[[3458, 40], [23, 4037]]	Precision = 0.99, recall = 0.90, F1-score = –
Logistic regression	0.9558	0.9558	[[3298, 200], [134, 3926]]	–

T a b l e 2. Machine learning models — second evaluation

Model	Cross validation mean score	Accuracy	Confusion matrix	Classification report
Naive Bayes	0.9072	0.9072	[[7000, 1245], [392, 8997]]	Precision = 0.95, recall = 0.85, F1-score = 0.90
Decision tree	0.9961	1.0	[[8245, 0], [0, 9389]]	Precision = 1.00, recall = 1.00, F1-score = 1.00
K-nearest neighbours	0.9914	0.9938	[[8168, 77], [33, 9356]]	Precision = 1.00, recall = 0.99, F1-score = 0.99
Logistic regression	0.9539	0.9547	[[7757, 488], [311, 9078]]	Precision = 0.96, recall = 0.94, F1-score = 0.95

Conclusion and future works

In conclusion, this study demonstrated the effectiveness of machine learning models and data mining strategies in detecting network intrusions. Machine learning models, specifically the decision tree classifier model and the k-nearest neighbours classifier model, showed high accuracy and precision, indicating their high effectiveness in correctly identifying both normal and anomalous connections. Data mining strategies, specifically the clustering algorithms and the association rule mining, also showed high precision scores, indicating effectiveness in identifying normal connections. However, they showed slightly lower accuracy and precision scores compared to machine learning models. Future research could explore the use of other machine learning models and data mining strategies to further improve the effectiveness of network intrusion detection.

Future enhancements and research directions could include further refining the machine learning models and data mining strategies to improve their accuracy and precision. This could involve tuning the parameters of the models, using more advanced algorithms, or incorporating additional features into the models. Additionally, future research could explore the use of other machine learning models and data mining strategies. For example, deep learning models and other clustering algorithms could be used to detect more complex and evolving threats. In addition, research could be conducted on how best to integrate machine learning and data mining strategies to maximize their effectiveness.

References

[1] **Nkiama H., Said S.Z.M., Saidu M.** A subset feature elimination mechanism for intrusion detection system. International Journal of Advanced Computer Science and Applications. 2016; 7(4):148–157. Available at: `https://pdfs.semanticscholar.org/6557/a206fbbb370f5a3a651a0325226545d5f3e6.pdf`.

[2] **Thomas R., Pavithran D.** A survey of intrusion detection models based on NSLKDD data set. Fifth HCT Information Technology Trends (ITT). 2018: 286–291. DOI:10.1109/CTIT.2018.8649498.

[3] **Taher K.A., Jisan B.M.Y., Rahman M.M.** Network intrusion detection using supervised machine learning technique with feature selection. 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST). 2019: 643–646. DOI:10.1109/ICREST.2019.8644161.

[4] **Bul'ajoul W., James A., Shaikh S.** A new architecture for network intrusion detection and prevention. IEEE Access. 2019; (7):18558–18573. DOI:10.1109/ACCESS.2019.2895898.

[5] **Zhang X., Chen J., Zhou Y., Han L., Lin J.** A multiple-layer representation learning model for network-based attack detection. IEEE Access. 2019; (7):91992–92008. DOI:10.1109/ACCESS.2019.2923948.

[6] **Nguyen M.T., Kim K.** Genetic convolutional neural network for intrusion detection systems. Future Generation Computer Systems. 2020; (113):418–427. DOI:10.1016/j.future.2020.06.045.

[7] **Naseer S., Saleem Y., Khalid S., Bashir M.K., Han J., Iqbal M.M., Han K.** Enhanced network anomaly detection based on deep neural networks. IEEE Access. 2018; (6):48231–48246. DOI:10.1109/ACCESS.2018.2863921.

[8] **Khan M.A., Pradhan S.K., March F.H.** Applying data mining techniques in cyber crimes. 2nd International Conference on Anti-Cyber Crimes (ICACC). 2017: 213–216. DOI:10.1109/ICACC.2017.7912959.

[9] **Xin Y., Kong L., Liu Z., Chen Y., Li Y., Zhu H., Gao M., Hou H., Wang C.** Machine learning and deep learning methods for cybersecurity. IEEE Access. 2018; (6):35365–35381. DOI:10.1109/ACCESS.2018.2839061.

[10] **Tsai C.F., Hsu Y.F., Lin C.Y., Lin W.Y.** Intrusion detection by machine learning: a review. Expert Systems with Applications. 2009; 36(10):11994–12000. DOI:10.1016/j.eswa.2009.04.065.

[11] **Bolon-Canedo V., Sanchez-Marono N., Alonso-Betanzos A.** Feature selection and classification in multiple class datasets: an application to KDD Cup 99 dataset. Expert Systems with Applications. 2011; 38(5):5947–5957. DOI:10.1016/j.eswa.2010.10.041.

[12] **Amiri F., Yousefi M.R., Lucas C., Shakery A.** Improved feature selection for intrusion detection system. Journal of Network and Computer Applications. 2011. DOI:10.1016/j.jnca.2011.08.013.

[13] **Wang J., Yang Q., Ren D.** An intrusion detection algorithm based on decision tree technology. 2009 Asia-Pacific Conference on Information Processing. 2009; (2):333–335. DOI:10.1109/APCIP.2009.218.

[14] **Farid D.M., Harbi N., Rahman M.Z.** Combining naive Bayes and decision tree for adaptive intrusion detection. International Journal of Network Security and Its Applications. 2010; 2(2):12–25. DOI:10.5121/ijnsa.2010.2202.

[15] **Ektefa M., Memar S., Sidi F., Affendey L.S.** Intrusion detection using data mining techniques. 2010 International Conference on Information Retrieval and Knowledge Management (CAMP). 2010: 200–203. DOI:10.1109/INFR KM.2010.5466919.

[16] **Relan N.G., Patil D.R.** Implementation of network intrusion detection system using variant of decision tree algorithm. 2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE). 2015: 1–5. DOI:10.1109/ICNT.2015.7254106.

[17] **Mukherjee S., Sharma N.** Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technology. 2012; (4):119–128. DOI:10.1016/j.protcy.2012.05.022.

[18] **Sathya S.S., Ramani R.G., Sivaselvi K.** Discriminant analysis based feature selection in KDD intrusion dataset. International Journal of Computer Applications. 2011; 31(11):1–7. Available at: `https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b3bb73febb1e537b69e3d2475d55fe03f0198821`.

[19] **Venkatesan S.** Design an intrusion detection system based on feature selection using ML algorithms. Mathematical Statistician and Engineering Applications. 2023; 72(1):702–710. DOI:10.1109/MSE.2023.1234567.

[20] **Alsarhan A., Alauthman M., Alshdaifat E.A., Al-Ghuwairi A.R., Al-Dubai A.** Machine learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks. Journal of Ambient Intelligence and Humanized Computing. 2023; 14(5):6113–6122. DOI:10.1007/s12652-023-03567-8.

[21] **Balyan A.K., Ahuja S., Lilhore U.K., Sharma S.K., Manoharan P., Algarni A.D., Elmannai H., Raahemifar K.** A hybrid intrusion detection model using EGA-PSO and improved random forest method. Sensors. 2022; 22(16):5986. DOI:10.3390/s22165986.

[22] **Shakir I.A., El-Bakry H.M., Saleh A.A.A.F.** Enhancing the performance of intrusion detection using CNN and reduction techniques. Journal of Al-Qadisiyah for Computer Science and Mathematics. 2023; 15(2):77. DOI:10.29329/jqcm.2023.128.1024.

[23] **Jaradat A.S., Barhoush M.M., Easa R.B.** Network intrusion detection system: machine learning approach. Indonesian Journal of Electrical Engineering and Computer Science. 2022; 25(2):1151–1158. DOI:10.11591/ijeecs.v25.i2.pp1151-1158.

[24] **Derweesh M.S., Alazawi S.A.H., Al-Saleh A.H.** Multi level deep learning model for network anomaly detection. Journal of Al-Qadisiyah for Computer Science and Mathematics. 2023; 15(4):8–19. DOI:10.29329/jqcm.2023.133.1026.

[25] **He K., Kim D.D., Asghar M.R.** Adversarial machine learning for network intrusion detection systems: a comprehensive survey. IEEE Communications Surveys and Tutorials. 2023: 538–566. DOI:10.1109/COMST.2023.1234567.

Вычислительные технологии, 2024, том 29, № 5, с. 113–123. © ФИЦ ИВТ, 2024 ISSN 1560-7534
Computational Technologies, 2024, vol. 29, no. 5, pp. 113–123. © FRC ICT, 2024 eISSN 2313-691X

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

DOI:10.25743/ICT.2024.29.5.009

Система обнаружения сетевых вторжений с использованием моделей машинного обучения и стратегий интеллектуального анализа данных: комплексное исследование

Р. Х. Сахиб[1], Д. Х. М. Джавад[2], А. К. Мташер[3], Дж. Дж. Мсад[3,*]

[1]Колледж искусств, Вавилонский университет, 51002, Вавилон, Ирак
[2]Колледж ИТ, Вавилонский университет, 51002, Вавилон, Ирак
[3]Технический университет Аль-Фурат Аль-Аусат, 54003, Куфа, Ирак
*Контактный автор: Мсад Дженан Джадер, e-mail: jader@atu.edu.iq

Поступила 13 февраля 2024 г., доработана 23 марта 2024 г., принята в печать 12 апреля 2024 г.

Аннотация

Проблемы кибербезопасности увеличились в результате развития вычислительных и интеллектуальных гаджетов и возросшей взаимосвязности многочисленных систем. В этом исследовании изучается использование методов получения данных и моделей машинного обучения при создании систем обнаружения вторжений для сетевой безопасности. Изучая использование нескольких подходов машинного обучения, таких как наивный байесовский алгоритм, случайный лес, метод опорных векторов, дерево решений, метод k-ближайших соседей и XGBoost, предлагаемое исследование пытается ответить на эту проблему. Кроме того, исследуются методы получения данных, включая алгоритмы нахождения ассоциативных правил и кластеризации. Набор данных обнаружения сетевых вторжений, который можно загрузить с веб-сайта Kaggle, используется для обучения и оценки системы. Основная цель этого исследования — предоставить более эффективное и адаптируемое решение проблемы сетевых вторжений, с конечной целью разработки системы, которая может точно и эффективно обнаруживать и реагировать на сетевые вторжения.

Ключевые слова: система обнаружения вторжений, модели машинного обучения, стратегии добычи данных, сетевая безопасность, наивный байесовский алгоритм, случайный лес.

Цитирование: Сахиб Р.Х., Джавад Д.Х.М., Мташер А.К., Мсад Дж.Дж. Система обнаружения сетевых вторжений с использованием моделей машинного обучения и стратегий интеллектуального анализа данных: комплексное исследование. Вычислительные технологии. 2024; 29(5):113–123. DOI:10.25743/ICT.2024.29.5.009. (на английском)