European Journal of Research Development and Sustainability (EJRDS)
Available Online at: https://www.scholarzest.com
Vol. 5 No 01, January 2024
ISSN: 2660-5570

WIRELESS INTRUSION DETECTION SYSTEM-BASED ON NEURAL NETWORK APPROACHES

Ammar Ali Mohammed Ridha
Information Systems department, Directorate of Communication and Information Technology, Ministry of Interior, Baghdad, Iraq
Email: ammar.intelligent@gmail.com

Article history:

Received: 28th November 2023
Accepted: 26th December 2023
Published: 30th January 2024

Abstract:

Wireless network security is crucial to privacy, especially when sensitive data is exchanged across it. Most systems now use modern online services for government, banking, email, and marketing, making this issue more important. After completing the preparatory processing of the chosen data set, numerous methods were utilized to reduce the total number of features in forming the neural network classifier. In addition to training and testing the system, we also examined the accuracy of the classifier and the detection and error rates based on the Matlab program. The findings of practical experiments indicate that the performance improved when relying on the data set with reduced features rather than the data set with full features. This resulted in improved Detection rates and low positive error rates for all five offensive classes in addition to the natural class. The system has attained an average level of accuracy in determining what constitutes offensive communication and what constitutes normal conversation % 99.7. The system has been evaluated using the KDDCup99 standard data set, which is widely used in the field of intrusion detection research by a large number of researchers as well as others who are interested in this field.

Keywords: Intrusion detection system, neural networks, dataset, KDDCup99 preprocessing, feature reduction.

INTRODUCTION

The strong demand for internet use is continuously expanding, and along with it comes an increase in the number of dangers posed to the network. According to research that was conducted by Symantec in 2019 and released in 2019, the business identified more than 430 million new forms of dangerous software in 2016, which is an increase of 36 percent greater than the previous year [1]. assaults can take many different forms across a wide range, including Brute Force Attacks, Heartbleed Attacks, DoS assaults, DDoS attacks, Web attacks, and many others. The expansion of the network's bandwidth is precisely proportional to the rate at which the number of people using the internet is consistently going up, which is a very encouraging sign. There is a significant gap between the usual rate of 1 Gigabit per second and 10 Gigabits per second for the ordinary data center in today's world. Companies on the cutting edge of technology, such as Google, Facebook, and others, as well as large corporations, experience speeds ranging from 40 to 100 gigabits per second (Gbps) when downloading and uploading data [1, 2]. A Network-based Intrusion Detection System (IDS) is a safety technology that protects against attacks from within the network, outside the web, and unauthorized network access. Either software or hardware can develop IDS. Most people are familiar with the idea of a firewall, constructed to protect a whole network from unwanted access based on IP address and port number. IDS is responsible for controlling these types of operations. The monitoring of network traffic allows for the identification of the number of intrusion attempts, such as hacking activities or denial-of-service attacks, which could jeopardize the security of any individual computer or the security of the entire network [2]. The wireless intrusion detection system (WIDS) is typically installed outside of the firewall, which enables monitoring of all outbound traffic through the sensing and detection of abnormal behaviors.

RELATED WORK

Researchers worldwide have long been keen on studying network security topics, particularly intrusion detection systems (IDS). Many researchers rely on the KDDcup99 dataset for evaluating various IDSs, employing a variety of methods and processes. For example, Subarna and Kaphle [3] introduced a novel learning methodology for developing an innovative IDS using backpropagation neural networks (BPN) and self-organizing maps (SOM), comparing their performance. The IDS's primary function is to protect resources from threats by analyzing and predicting user behavior, distinguishing between attacks and normal activities. The proposed strategy significantly reduces training time and yields positive training results, offering a powerful tool for investigating, simulating, and understanding the complex attack behaviors associated with electronic crime. Akinwande et al. [4] are constructed using Artificial Immune System (AIS) algorithms, particularly AIRS1, Immunos, Dendritic Cell and ClonalG Algorithms. These algorithms were evaluated using the CICIDS 2017 and NSL-KDD datasets, which provide a public, up-to-date dataset for Botnets and malicious attacks, including DDoS, XSS, SQL Injection, Infiltration, Bruteforce, and Portscan. Their results demonstrate that AIS algorithms are a new approach for IDS that detect new attacks more accurately and precisely than other classifiers. Deep neural network intrusion detection was proposed by Gowdhaman and Dhanapal [5]. Cross-correlation is used to choose the dataset's best features. A deep neural network detects intrusions using the provided parameters. The experimental results showed that the suggested DNN outperforms support vector machines, decision trees, and random forest in identifying assaults. Hui et al. [6] proposed a revolutionary Convolutional Neural Network-based intrusion detection approach (CNN). IDS-CNN, an efficient, real-time, and automated intrusion detection system, is constructed using the proposed approach. They build their system using Tcpdump, Bro, and Tensorflow, open-source tools. Data preprocessing, neural network training, network testing, and intrusion response are implemented on Linux OS. Finally, the simulation experiment with the NSL-KDD data set and the actual network flow test show that the proposed IDS-CNN system can efficiently detect network data stream intrusions and has greater precision than the state-of-the-art method. Adeel et al. [7] devised an ensemble-based intrusion detection model, integrating logistic regression, naive Bayes, and decision trees with a voting classifier after evaluating several state-of-the-art approaches. They utilized the CICIDS 2017 dataset, achieving significant accuracy improvements in both binary and multi-class classification scenarios compared to existing models. Amar et al. [8] examined extreme network methods' detection concerning power level and node momentum for two mobility models: Random waypoint (RWP) and Gauss Markov (GM). They identified malicious activities like black holes and DDoS attacks, achieving detection rates over 98% in high power/node velocity scenarios but dropping to around 90% in low power/node velocity scenarios. Sharuka et al. [9] trained a Deep Neural Network (DNN) on 28 NSL-KDD features, implementing a machine learning pipeline with categorical data encoding and feature scaling. They utilized real-time data extracted by a C++ feature extractor between the gateway router and LAN, achieving 81% accuracy, 96% precision, 70% recall, and 81% f1-score. The study elucidates system implementation and operation, offering insights into building advanced IDSs capable of detecting modern intrusions. Ansam Khraisat [10] proposed a contemporary IDS arrangement, providing a comprehensive analysis of recent research and commonly used evaluation datasets. The study also addresses future research challenges to enhance computer system security, along with evasion strategies used by attackers to evade detection.

Proposed Methods

1- Overview

This section outlines the process of developing a neural network classifier for network intrusion detection, primarily using the KDD99 dataset. It begins with preliminary data processing, emphasizing the need for efficient data preparation due to the large volume of data involved. The initial structure of the neural network classifier is constructed using all attributes for comparison purposes. Multiple techniques are then employed to select the most relevant attributes for data traffic classification, aiming to reduce the number of attributes required for intrusion detection while maintaining effectiveness. A model is built based on the processed data to swiftly and effectively detect intrusions. The text presents the conclusive structure of the neural network classifier and compares it with prior research. MATLAB R2014a is utilized for software implementation, experimentation, and performance evaluation of the classifier. After each experiment, the classifier's performance is assessed to refine its structure and achieve optimal performance. A block diagram of the implemented system is provided in Figure 4.1, illustrating the architecture of the developed neural network classifier for network intrusion detection.

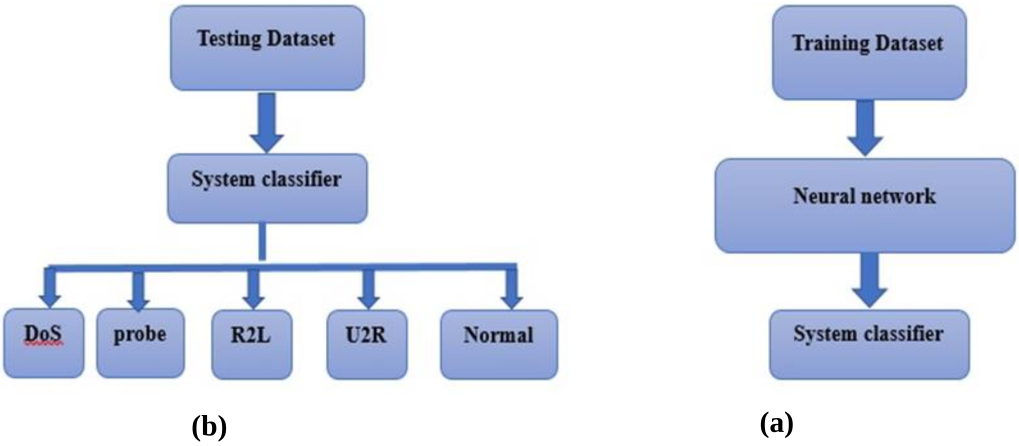

Figure 1 System diagram

The phases of designing the system classifier using the processed training data set are depicted in Figure (a). Then, using the test data set that will be classified as offensive log or natural, evaluate it as shown in Figure (b). This chapter explains the tools and methodologies utilized to conduct the research, as well as their placement within the research.

2- Data cleaning and Preprocessing

Because the data are redundant and the attributes are not important, there is a requirement for preparatory processing of the data, which can be observed from the fact that there is a need. The categorization method is frequently cluttered, which leads to cognitive detection that is erroneous or otherwise failed. Additionally, there are when all of the features are used, the processing time will take longer. Finally, preprocessing is beneficial because it allows for the elimination of superfluous data, the completion of any missing data, and the standardization of the data's format. The following activities were carried out at this stage:

- Delete duplicate records.
- Convert symbolic data into numeric values
- Feature Reduction

3- Proposed System architecture

MATLAB was used to train and save networks using the backpropagation algorithm, which were subsequently applied and evaluated using a program shown in the Appendix; figure (4.4) demonstrates MATLAB was used to conduct practical experiments that depended on varying the number of features involved in the classification process. The success of these investigations depended on the number of features utilized in the classification procedure. A box diagram outlining the software's involved procedures. The input and output files are examples of inputs that the software takes in. The input file is a collection of test data, and the target file is where the results of the classification of each record in the input file are stored. This allows for a comparison between the classification result produced by the system, and the actual classification result, i.e., labels. The input file is sent to the neural network classifier so that, based on what the neural network was previously trained on, each of these records may be classified.

The software evaluates the system's performance by comparing its classification result with the target file. If they match, the software displays "TRUE"; otherwise, it shows "FALSE". If a record is misclassified, it is flagged as inaccurate. For accurate classifications, the software continues to display the record's class type, which could be an intrusion type like "U2R" or "R2L", or a normal type like "DoS", "Probe", or "No Intrusion Detected".

The software depends on the results of the previous classification to calculate a set of results that are represented by the number of "Total Mismatched Cases," as well as the number of classification cases that are broken down into detail as follows: the number of cases that were correctly identified as intrusive, also known as the number of cases that were correctly classed as positive "Number of True Positive Cases (TP)" "Number of True Negative Cases," "Number of True Negative Cases," "Number of True Negative Cases," "Number of True Negative Cases," The "Number of False Positive Cases," the "Number of False Positive Cases," and the "Number of False Positive Cases" FP refers to the total amount of incorrect negative results, also known as done. Classify them as regular instances, but in reality, they are infiltrative cases, referred to as the "number of false negative cases" (TN). The averages of the previous numbers will then be displayed.

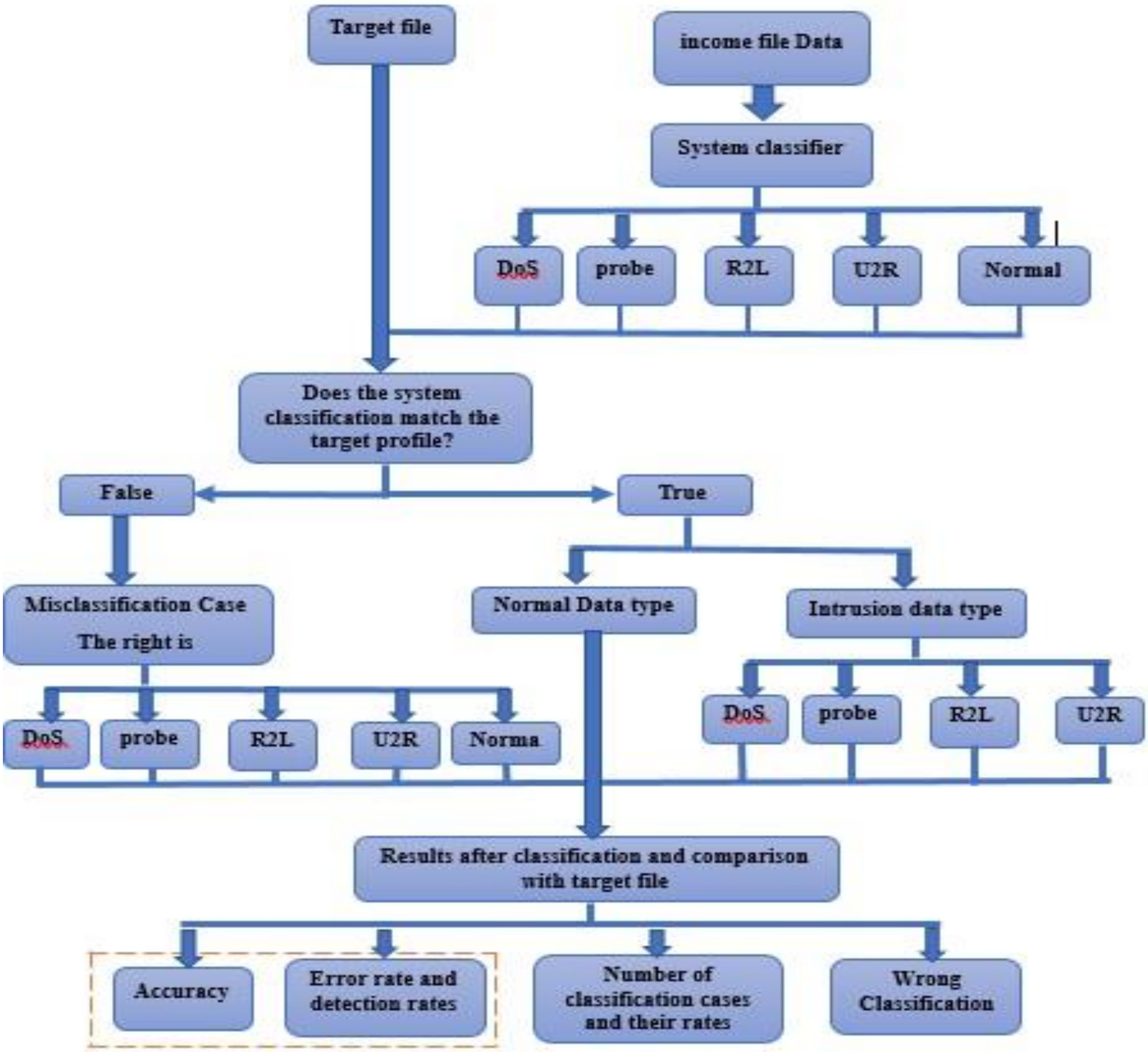

Figure 2 Evaluation modeling system architecture

Experimental Design

In this section, we will introduce the training and assessment test that will be implemented in our proposed method.

1- Training Dataset

The "10% KDD" dataset, derived from the larger KDD dataset, is widely used for training intrusion detection systems (IDS). It is a condensed version of the complete KDD dataset, containing 494,021 connection records. Notably, the distribution of attack types in the dataset is uneven, with denial-of-service attacks comprising the majority of communication records. The dataset encompasses 22 different types of attacks, with offensive communication records outnumbering natural ones. Additionally, it contains more instances of offensive communication records than natural ones.

2- Testing Dataset

For the purposes of testing, the dataset known as "Corrected KDD" is utilized. This collection gives data with a different statistical distribution when compared to the training dataset, regardless of whether it is "10% partial KDD" or complete KDD, as shown in Table (5.2) for the data set "311029." A KDD that has been corrected will have a contact record that is either designated as normal or as one of the four different attack kinds. The ratings predictions that were produced through testing are checked for accuracy by using the addresses of the connection records.

RESULTS

Agile testing and test automation are also current popular topics. Less frequently is it addressed, however, how to manage the (automated or manual) data required for software testing. The success or failure of software development and testing would depend on meticulously prepared data cases. You can't use just some data or just a random test case. You will need a high-quality and representative data set to test a software application effectively. The optimal test set identifies all application errors using the minimal data set possible in summary, which requires a small (test) data set that is realistic, reliable, and flexible. The program will evaluate the system's effectiveness by contrasting the categorization result produced by the system with the target file. Therefore, the result that is displayed by the program is either a true classification, denoted by the word "TRUE," if the classification result matches the file that is being targeted, or a false classification, denoted by the word "False," if the system classification result is not matching the target file. Following the findings of the comparison with the "Misclassified Case" target file, the examined record is inaccurate. In the event that the classification result is accurate, the program will continue to display the class type to which the investigated record belongs. This may be a message that reads, "Data type of this vector is Intrusion type," or it could be a natural record that reads "U2R" or "R2L."

Table 1 illustrates the results of classifying a record batch.

DOS class		**Probe class**	
False negative	1.630220e-03	**False negative**	5.260341e-03
False positive	5.260341e-03	**False positive**	1.630220e-03
True positive	9.947397e-01	**True positive**	9.983698e-01
True negative	9.983698e-01	**True negative**	9.947397e-01

So, here are the Accuracy of the system: **99.7**

CONCLUSION

The paper discussed the prevalent use of the KDD99 dataset in intrusion detection systems (IDS). It highlights a study that aimed to improve IDS performance using the back-propagation method of neural networks. MATLAB was employed to train the neural network and present detailed results for each tested record, distinguishing between attack types (DoS, Probe, R2L, U2R) and natural records. System accuracy was determined based on correct classifications and detection rates. The text suggests future research directions, urging the exploration of other datasets and the introduction of new attack types while maintaining the neural network with the backpropagation algorithm as a benchmark for comparison.

REFERENCE

1- A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, pp. 1-22, 2019.

2- H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16-24, 2013.

3- N. Shabani and A. Munir, "A review of cyber security issues in hospitality industry," in Intelligent Computing: Proceedings of the 2020 Computing Conference, Volume 3, 2020, pp. 482-493: Springer.

4- S. Shakya and B. R. Kaphle, "Intrusion detection system using back propagation algorithm and compare its performance with self organizing map," Journal of Advanced College of Engineering and Management, vol. 1, pp. 127-138, 2015.

5- O. T. Akinwande and M. B. Abdullahi, "Performance Evaluation of Artificial Immune System Algorithms for Intrusion Detection," 2019.

6- V. Gowdhaman and R. Dhanapal, "An intrusion detection system for wireless sensor networks using deep neural network," Soft Computing, pp. 1-9, 2021.

7- H. Wang, Z. Cao, and B. Hong, "A network intrusion detection system based on convolutional neural network," Journal of Intelligent & Fuzzy Systems, vol. 38, no. 6, pp. 7623-7637, 2020.

8- A. Abbas, M. A. Khan, S. Latif, M. Ajaz, A. A. Shah, and J. Ahmad, "A new ensemble-based intrusion detection system for internet of things," Arabian Journal for Science and Engineering, pp. 1-15, 2021.

9- A. Amouri, V. T. Alaparthy, and S. D. Morgera, "A machine learning based intrusion detection system for mobile Internet of Things," Sensors, vol. 20, no. 2, p. 461, 2020.

10- S. P. Thirimanne, L. Jayawardana, L. Yasakethu, P. Liyanaarachchi, and C. Hewage, "Deep neural network based real-time intrusion detection system," SN Computer Science, vol. 3, no. 2, p. 145, 2022.