Journal of Education and Science (ISSN 1812-125X), Vol: 31, No: 03, 2022 (99-109)

Proposing a Model for Detecting Intrusion Network Attacks Using Machine Learning Techniques

Teba Ali Jasem[1*], Muna M.T. Jawhar [2]

[1*,2]Softwares Department, College of Computer Sciences & Mathematics, University of Mosul, Mosul, Iraq

E-mail: [1*]tebaa.20csp6@student.uomosul.edu.iq, [2]dr.muna_taher@uomosul.edu.iq

(Received May 15, 2022; Accepted August 02, 2022; Available online September 01, 2022)

DOI: 10.33899/edusj.2022.133867.1240,

Abstract: At the present time, the reliance on computers is increasing in all aspects of life, so it is necessary to protect computer networks and computing resources from complex attacks against the network. This is performed by building tools, applications, and systems that detect attacks or anomalies adapting to ever-changing architectures and dynamically changing threats. The goal of this paper is to build a Network Intrusion Detection System (NIDS) based on deep learning techniques such as Convolutional Neural Network (CNN), which demonstrated its efficiency in predicting, classifying, and extracting high-level features in network traffic.

Keyword: Network security, Deep Learning, IDS, KDD Dataset, Convolutional Neural Network.

مقترح نموذجً لاكتشاف هجمات تسلل الشبكة باستخدام تقنيات التعلم الآلي

طيبة علي جاسم[1*]، منى محمد طاهر جوهر[2]

[2،*1] قسم البرمجيات، كلية علوم الحاسوب والرياضيات، جامعة الموصل، الموصل، العرا ق

الملخص: في الوقت الحاضر ، يتزايد الاعتماد على أجهزة الكمبيوتر في جميع جوانب الحياة ، لذلك من الضروري حماية شبكات الكمبيوتر وموارد الحوسبة من الهجمات المعقدة ضد الشبكة. يتم ذلك عن طريق بناء الأدوات والتطبيقات والأنظمة التي تكتشف الهجمات أو الحالات الشاذة التي تتكيف مع البنى المتغيرة باستمرار والتهديدات المتغيرة ديناميكيًا. الهدف من هذه الورقة هو بناء نظام اكتشاف اختراق الشبكة (**NIDS**)استنادًا إلى تقنيات التعلم العميق مثل الشبكة العصبية التلافيفية (**CNN**) ، والتي أظهرت كفاءتها في التنبؤ والتصنيف واستخراج الميزات عالية المستوى في حركة مرور الشبكة.

الكلمة المفتاحية: أمان الشبكة ، التعلم العميق ، **IDS**، مجموعة بيانات **KDD** ، الشبكة العصبية التلافيفية.

Introduction

Providing network security is one of the most important things in network communications, considering network growth and the increase of devices being added to the network, a fact that demands more requirements to provide network security. A network security system is necessary to protect devices and data belonging to network users, and it also helps protect information shared on the network, protect people's personal information, and helps prevent users from falling victim to pirates [1].

Network security technologies must be constantly developed. Network traffic attack detection systems play an important role in network security by detecting intrusions and the competent authority is alerted. There are two types of attack detection systems on the network: flaw detection and abuse detection [2]. In detecting anomalies and detecting defects the database of normal activity is identified and any deviations from the normal activity are alerted to the occurrence of intrusion or attacks in the network. Abuse detection detects the sorts of attacks in the database and if the same types of possibilities exists in the network, after which they are classified as attacks or hacks [3]. Several AI methods, such as rule-based and data mining approaches, have been developed for an intrusion detection system, and it is the first suggested framework for IDSs. It is a way of gathering information from a large-scale database that aids in the extraction of patterns from the knowledge base, IDS, and the use of this knowledge to forecast the occurrence of intrusion. [4]. In these types of systems, there are drawbacks that do not have the ability to detect new attacks, so many machine learning (ML) algorithms have been developed to promote AIDS and they are MLAIDS [5], these algorithms assess the network's health by categorizing and processing data as normal or abnormal.

The second section in this paper includes the previous works of researchers in the same field. The third section explains the data used. The fourth section includes the methodology. The fifth section includes the results and discussion, and the last section includes conclusion.

The main objective of the paper is to design a software tool or system that detects intrusions or attacks across the network using machine learning techniques. We also aim to improve the system over time through automatic learning and using the latest deep learning algorithms to discover new attacks.

1. Previous works

Because of the wide use of computer networks in various fields of life, network security including intrusion detection systems, is a source of interest for many researchers. In 1998 and 1999 Cup was applied in 42% and 20% of these studies respectively [6]. Since 1999, with the emergence of different algorithms and techniques used in the field of intrusion detection systems, many researchers have used the KDD Cup, which is the most widely used dataset in studies of network-based intrusion detection systems [7]. In 2019, Y. Xiao et al. [8] used Batch Normalization with the KDD99 Dataset to construct a CNN-based IDS using an auto-encoder (AE) network as a dimensionality reductional technique. The suggested framework also deleted unneeded and superfluous features to monitor network and host level activities. The authors of [9] present two approaches for detecting DDoS attacks on SDN. SVM and Deep Neural Network (DNN) approaches were then utilized to categorize the attack in the first phase, which used a signature-based anomaly detection system for network data. The KDDCUP99 dataset was then utilized for training and detection by the authors. The trial findings showed that DNN outperforms SVM, with accuracy rates of 92.30% and 74.30% respectively. Vinaya-kumar et al. [10] developed a hybrid IDS. DNN outperformed other standard machine learning classifiers after an exhaustive comparison study using several machine learning and deep learning classifiers. Yin and colleagues [11] used RNN the DL algorithm, the input to a network is 122-dimensional neurons, when tested on the NSL-KD data set, their model achieved an

accuracy score of 83.28% in binary classification and 81.29% in multiple classifications. Cui Zihua et al. code in pictures for classification using CNN implemented with multiple malware image sizes (24*24, 48*48, 96*96, 192*192) and CNN. Riyaz et al. [12] used the KDD-99 dataset to create an IDS using a CNN architecture for use in wireless networks. A unique coefficient-based feature selection technique (CRF-LCFS) was used in the framework, which improved the model's detection accuracy and computation times. The proposed method had a detection accuracy of 98.9% and a false alarm rate of less than 1%, according to the study.

2. The dataset

In this paper, we used the standard datasets of the KDD Cup "99". DARPA generated the KDD'99 dataset in 1999, based on network traffic from the 1998 dataset. For each network connection, it is running 41 Pre-processed features. There are four groups of features in the KDD'99 data collection.

Core features (#1 to #9), content features (#10 to #22), time-based traffic features (#23 to #31), and host depend on Traffic features (#32 to #41) are just a few examples. KDD'99 [13] has a total of 4,898,430 records. They are used to create traffic by spoofing different IP addresses. The traffic characteristics are then recorded in TCP dump format. A total of seven weeks was allotted for simulation. Normal connections to the supposed IP are established in a military network. There are various attacks, each class containing 21 kinds of properties [14], which fall under four types of attacks: “DOS attacks, Probe attacks, R2L attacks and U2R attacks” [15-16]. The imitated attacks were divided into four categories [17]:

1. Denial of Service Attack (DoS): When an attacker renders a computer or memory resource too busy or full to process valid requests, or refuses legitimate users access to a system, this is known as a DoS attack.
2. User to Root Attack (U2R): This is a type of exploit in which the attacker gains access to a regular user account on the system (perhaps through password sniffing, a dictionary attack, or social engineering) and then exploits a vulnerability to get root access.
3. Remote to Local Attack (R2L): An attacker who has the capacity to transmit packets to a machine across a network but does not have an account on that machine exploits a vulnerability to get local access as a user of that system is known as a remote to local attack (R2L)[18]
4. Probing Attack: is an attempt to obtain information about a network of computers in order to obfuscate the network's security mechanisms [17]

3. Convolutional Neural Networks CNN

The technique of data representation is crucial to the success of machine learning algorithms [19]. Many researchers, such as Niaz et al., have employed deep neural networks to produce an effective intrusion detection system [18], and these studies build their models to learn represent manually chosen traffic characteristics, rather than fully utilizing deep neural network capabilities [20]. As learning characteristics must be retrieved directly from the raw data, as in computer vision and natural language processing [21], CNN is the most often used deep neural network architecture. CNN utilizes the original data as direct input to the network, does not need feature extraction or picture reconstruction, has a small number of parameters, and processes data. CNNs have been demonstrated to be extremely successful in image recognition and feature extraction [22].

The input layer, the hidden layer, and the output layer are the three fundamental components of a neural network. A CNN's hidden layer is composed of convolutional (non-linear) activation, aggregation (reduction), and fully linked layers. Weight sharing: the weights of connections between a subset of neurons are shared in the same layer, and during sampling, the convergence layer is regularly inserted

between succeeding convolutional layers. As a result, CNNs have a lower weight number than other neural networks, making them more efficient at detecting network invasion. In network intrusion detection, Vinayakumar et al. [23] demonstrate that CNN and its diverse structure outperform standard machine learning classifiers. According to the findings, one-dimensional convolution in CNN identifies network intrusions with high accuracy.

CNNs were first used in image processing applications as a biologically inspired deep learning model for image classification, face recognition, and pattern recognition [24-25]. Convolution operations use a "filter" "kernel" to automatically extract the complicated properties of patterns from a picture.

A CNN Network Architecture is made up of three types of layers:" convolutional layers, pooling layers, and fully connected layers ", see Figure 1. A CNN architecture is formed when these layers are grouped or layered. In addition to these three layers, the dropout layer and the activation function given below [26] are essential factors.

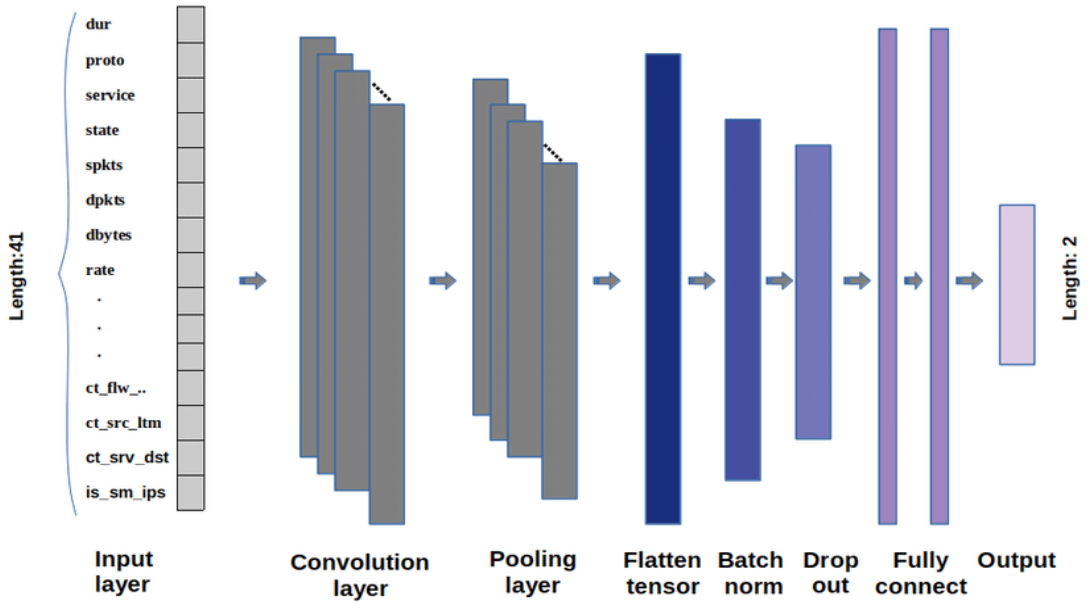

Figure.1 CNN layers

4. Methodology

4.1. Data pre-processing

The first part of our model is preparing the data, it divides the connections into two types: "attack" and "normal" categories based on the "labels" column. The "attack" category is then broken down into four main categories: DoS, Probe, R2L, and U2R. These classes are then indexed or coded, as we put all types of attacks into a single value, the "attack", to make the problem a binary classification.

The next step is data pre-processing in which each connection record has 41 features, 38 numeric characteristics, and 3 category features. The data set's category characteristics are encoded fields (0 to 2). They contain information such as the duration, protocol type, and service. Fields 3–40 include information on the connection's other characteristics, such as the number of unsuccessful logins, the number of files accessed, and the number of exit limits. The connection type is shown in the label column, which includes "normal" and "attack.", as shown in Figure 2, and Figure 3 illustrates the distribution of attacks in the dataset.

```
Out[76]: Index(['duration', 'protocol_type', 'service', 'flag', 'src_bytes',
       'dst_bytes', 'land', 'wrong_fragment', 'urgent', 'hot',
       'num_failed_logins', 'logged_in', 'num_compromised', 'root_shell',
       'su_attempted', 'num_root', 'num_file_creations', 'num_shells',
       'num_access_files', 'num_outbound_cmds', 'is_host_login',
       'is_guest_login', 'count', 'srv_count', 'serror_rate',
       'srv_serror_rate', 'rerror_rate', 'srv_rerror_rate', 'same_srv_rate',
       'diff_srv_rate', 'srv_diff_host_rate', 'dst_host_count',
       'dst_host_srv_count', 'dst_host_same_srv_rate',
       'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate',
       'dst_host_srv_diff_host_rate', 'dst_host_serror_rate',
       'dst_host_srv_serror_rate', 'dst_host_rerror_rate',
       'dst_host_srv_rerror_rate', 'label'],
      dtype='object')
```

Figure. 2 Characteristics of the data extracted from the data of KDD Cup "99"

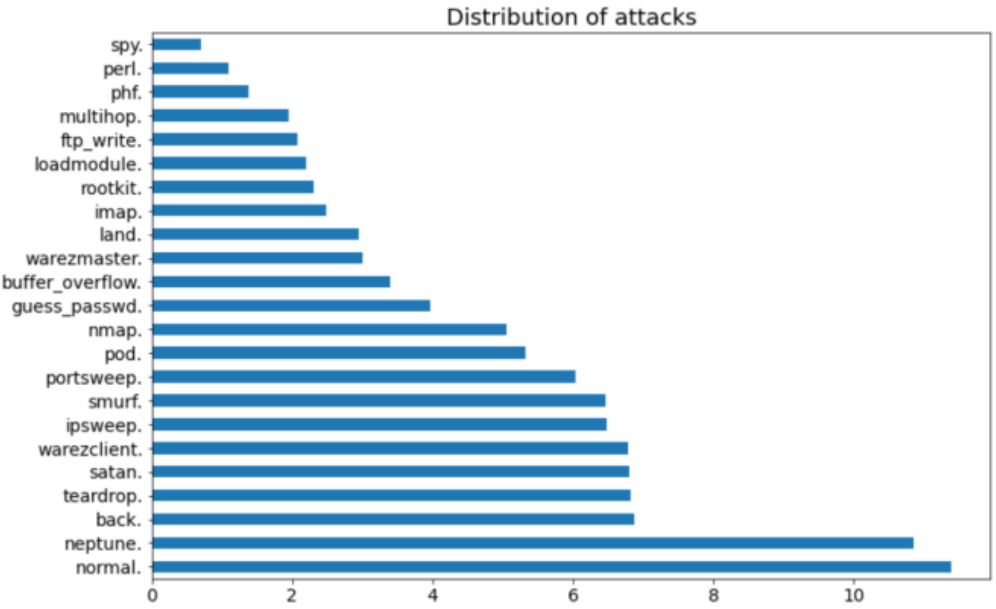

Figure. 3 Distribution of attacks in KDD Cup "99"

We employed a pre-processing procedure for this data in which we eliminated certain category elements and encoded others into numeric-only fields. These fields are sent into the CNN deep learning algorithm as inputs. Several tasks are included in this step:

1. Ensure that the data does not include any incorrect characters.

2. Remove any fields that have empty values or values that do not contain numbers.
3. Remove duplicate columns.

The main reason for pre-processing is because the data is in various forms and was gathered from various locations. It also assures the validity and efficacy of the model being trained on this data.

4.2. Feature selection

The first method for feature selection is to eliminate redundant and irrelevant data by fully selecting a subset of relevant features that represent the specific problem. The second method of feature selection is to have a more efficient form of classification, we took advantage of field correlation. In our model, there are several alternatives for picking the functionality, we can define many sets of interconnected variables and only leave one of them. Then we normalize or scale continuous values between all properties, so CNN trains the data in the same workspace, then all numerical features of the data set are normalized using z-scores.

4.3. Building the Model

The researchers used a CNN model with training data (80% of the data) using some base libraries such as Keras, Pandas, NumPy and Scikit-learn. We made a sequential CNN model with 3 layers. In the convolutional layer we put 64 nodes and kernels of size 3 and the activation function as relu, which is applied for each numeric value and replaces all negative values in the features with zero and the max pooling layer behind the convolutional layer. Although the max pooling layer is not required for a CNN model, we utilize it since there is very little chance of losing key features due to the max pooling as the converted data solely contains numerical information. Additionally, the flatten layer is followed by a dense layer and applied dropout with rate of 0.25. It is placed at the end of the network, which means that every time 75% of the nodes are randomly selected for a process, it will increase the randomness of the model and reduce the bias. Finally, there is a fully connected layer (dense layer) for output. Figure 4 shows a CNN architecture consisting of one convolutional layer, one pooling layer that produces the maximum value of two adjacent elements, two fully connected layers, an input layer followed by a dropout layer, an output layer.

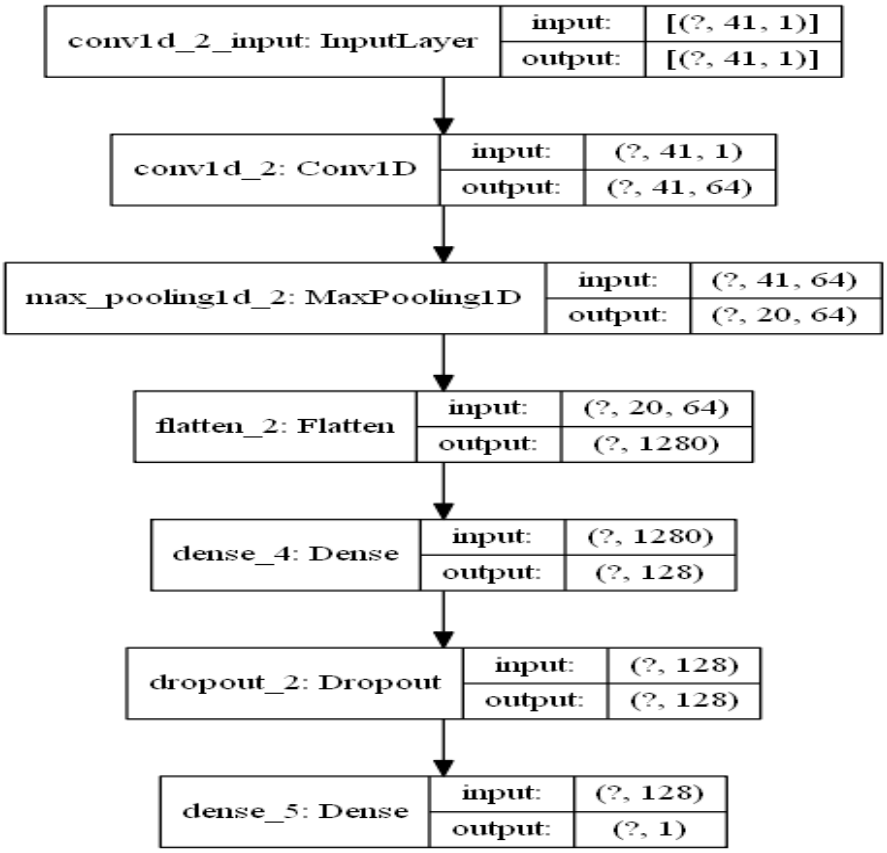

Figure 4: block diagram of the CNN architecture

5. Results and discussion

For our model, "ADAM" optimizer was used, batch size=128, epoch =30, while the activation function was "Sigmoid" for the output layer. Their summaries are given in Figure 5.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 41, 64)            256
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 20, 64)            0
_________________________________________________________________
flatten (Flatten)            (None, 1280)              0
_________________________________________________________________
dense (Dense)                (None, 128)               163968
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 164,353
Trainable params: 164,353
Non-trainable params: 0
```

Figure 5. The summaries of the model

The tested model was performed with test dataset, then the accuracy of the classifier was analysed as results and performance evaluation is shown in the following figures. The aim was to attain improved detection rates, classify types of attacks and determine the type of connection whether normal or abnormal. It reached an accuracy rate of 99.7%. distinguishing between offensive and normal contact.

Table 1. The accuracy and loss for each epoch

epoch	val_accuracy	val_loss	accuracy	Loss
0	0.979193	0.049888	0.972057	**0.046526**
1	0.970758	0.08094	0.972154	**0.046248**
2	0.969094	0.103922	0.97244	**0.045699**
3	0.971833	0.058877	0.972424	**0.045432**
4	0.969795	0.085976	0.972334	**0.045343**
5	0.967722	0.139465	0.972893	**0.044992**
6	0.972789	0.099025	0.972562	**0.044697**
7	0.982525	0.074684	0.97316	**0.044575**
8	0.967809	0.107305	0.972514	**0.044588**
9	0.967888	0.159589	0.972867	**0.044324**
10	0.97226	0.100143	0.972941	**0.044237**
11	0.978635	0.114767	0.972665	**0.044447**
12	0.972667	0.095729	0.972266	**0.044124**
13	0.966364	0.073877	0.972938	**0.043882**
14	0.984452	0.043519	0.972626	**0.043699**
15	0.975161	0.05451	0.972572	**0.043688**
16	0.974627	0.060094	0.972771	**0.043361**
17	0.971774	0.070533	0.972687	**0.043383**
18	0.970301	0.076172	0.97361	**0.04307**
19	0.975313	0.059245	0.973002	**0.043289**
20	0.969744	0.086635	0.973848	**0.042891**
21	0.977456	0.051748	0.973684	**0.042989**
22	0.983531	0.074282	0.973613	**0.043029**
23	0.98064	0.077685	0.973421	**0.042846**
24	0.974189	0.080109	0.974118	**0.042632**
25	0.979276	0.056337	0.973527	**0.042799**
26	0.979499	0.087185	0.973581	**0.04243**
27	0.975754	0.051565	0.973745	**0.042411**
28	0.974493	0.089369	0.973887	**0.042441**
29	0.986788	0.061391	0.973089	**0.042429**

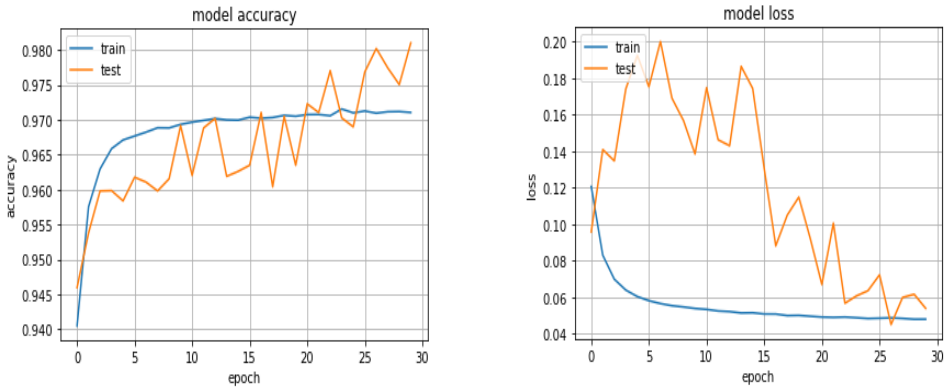

Figure 6. The result of accuracy and loss

To verify the results obtained through the proposed model, they were compared with the results of previous research, and through the comparison, we note that the results obtained from our model were higher than those of previous research, as shown in the following table.

Table 2. Comparison with previous research

Reference	Method	Dataset	Detection rate %	Classification
Javid et al. [10]	STL	NSL-KDD	88.38% 79.10 %	Binary Multi
Muhammad et al [12]	Genetic, Ranker, Greedy	NSL-KDD	81 77 77	/
Yin and colleagues [13]	RNN	NSL-KD	83.28 81.29	Binary Multi
Our research	CNN	KDDCUP99	98.10	Binary

6. Conclusions

The goal of the project was to build an IDS by using CNN that can classify the connections in the dataset as an "attack" or a" normal" connection. A binary classification model was built in which we used fully connected neural networks and convolutional neural networks to classify the types of connections and attacks. The system was trained and tested used KDD dataset, classifier accuracy and detection rates and error rates were measured using the PYTHON language and Jupiter Notebooks code editor. The results of practical experiments showed improved performance when using the low-attribute dataset, which was better than when using the full-attribute dataset. We reached an accuracy rate of 98.10% distinguishing between offensive and normal contact.

7. Acknowledgments

This work was supported by the University of Mosul / College of Computer Sciences and Mathematics. This work is part of the requirements for obtaining a master's degree.

8. References

[1] C. Wu and W. Li, "Enhancing intrusion detection with feature selection and neural network", *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3087-3105, 2021. Available: 10.1002/int.22397.

[2] R. Marwaha, "Intrusion Detection System Using Data Mining Techniques– A Review", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 7, no. 5, pp. 450-452, 2017. Available: 10.23956/ijarcsse/v7i5/0161.

[3] Mohammed, Maithem, and Ghadaa A. A. *"Network Intrusion Detection System Using Deep Neural Networks*." 1804.1 (2021). Print.

[4] I. Benmessahel, K. Xie and M. Chellal, "A new evolutionary neural networks based on intrusion detection systems using multiverse optimization", *Applied Intelligence*, vol. 48, no. 8, pp. 2315-2327, 2017. Available: 10.1007/s10489-017-1085-y.

[5] I. Ahmad, M. Basheri, M. Iqbal and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection", *IEEE Access*, vol. 6, pp. 33789-33795, 2018. Available: 10.1109/access.2018.2841987.

[6] S. Tamy, H. Belhadaoui, M. Rabbah, N. Rabbah and M. Rifi, "Select the Best Machine Learning Algorithms for Prediction and Classification of Intrusions using KDD99 Intrusion Detection Dataset", *Indian Journal of Science and Technology*, vol. 12, no. 37, pp. 1-6, 2019. Available: 10.17485/ijst/2019/v12i37/147551.yy

[7] Q. Niyaz, W. Sun and A. Javaid, "A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN)", *ICST Transactions on Security and Safety*, vol. 4, no. 12, p. 153515, 2017. Available: 10.4108/eai.28-12-2017. 153515..

[8] Xiao, Yihan, cheng xing, taining zhang, and zhongkai zhao. "An intrusion detection model based on feature reduction and convolutional neural networks." *IEEE Access* 7 (2019): 42210-42219.feature reduction and convolutional neural networks. IEEE Access, 7, 42210-42219.

[9] J. Mohammed, M. Hussain and U. Mirza, "Brachial plexus injury: Following birthday bumps", *Journal of Orthopaedics and Allied Sciences*, vol. 6, no. 2, p. 86, 2018. Available: 10.4103/joas.joas_23_18.

[10] C. Yin, Y. Zhu, J. Fei and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks", *IEEE Access*, vol. 5, pp. 21954-21961, 2017. Available: 10.1109/access.2017.2762418

[11] Z. Cui, F. Xue, X. Cai, Y. Cao, G. Wang and J. Chen, "Detection of Malicious Code Variants Based on Deep Learning", *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187-3196, 2018. Available: 10.1109/tii.2018.2822680..

[12] Vinayakumar, Ravi, Alazab, Mamoun, Soman, K. P., Poornachandran, Prabaharan, Al-Nemrat, A. and Venkatraman, Sitalakshmi. "Deep learning approach for intelligent intrusion detection system." *IEEE Access 7* (2019): 41525-41550.

[13] Choudhary, Sarika, and Nishtha Kesswani. "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT." Procedia Computer Science 167 (2020): 1561-1573.

[14] Güneş, H.; Kayacık, A.; and Nur Z. "Selecting Features for Intrusion Detection a Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets". *Conference on Privacy, Security and Trust. The Fairmont Algonquin*, 2005. St. Andrews, New Brunswick, Canada.

[15] L. Dhanabal, and S.P. Shantharajah, "A Study of NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms". *International Journal of Advanced Research in Computer and Communication Engineering*, 4, 446-452. - References - Scientific Research Publishing", Scirp.org, 2022. Available: https://www.scirp.org/reference/referencespapers.aspx?referenceid=2341209.

[16] S. El-Sappagh, A. Mohammed and T. AlSheshtawy, "CLASSIFICATION PROCEDURES FOR INTRUSION DETECTION BASED ON KDD CUP 99 DATA SET", *International Journal of Network Security & Its Applications*, vol. 11, no. 03, pp. 21-29, 2019. Available: 10.5121/ijnsa.2019.11302.

[17] S. K. Srivastava, Y. K. Sharma, and S. Kumar, "Characteristics categorization dataset KDD Cup'99," AIP Conf. Proc., vol. 2142, no. January 2020, 2019, doi: 10.1063/1.5122494.

[18] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *in 2009 IEEE symposium on computational intelligence for security and defense applications*, 2009, pp. 1–6.

[19] Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798-1828, 2013. Available: 10.1109/tpami.2013.50.

[20] M. Maithem and G. Al-sultany, "Network intrusion detection system using deep neural networks", *Journal of Physics: Conference Series*, vol. 1804, no. 1, p. 012138, 2021. Available: 10.1088/1742-6596/1804/1/012138.

[21] S. Eesa, Z. Orman, and A, Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems". *Expert Systems with Applications*. vol. 42, no. 5, pp. 2670–2679.View at: Publisher Site | Google Scholar (2015).

[22] J. Heaton, I. Goodfellow, Y. Bengio, and A. Courville", *Deep learning", Genetic Programming and Evolvable Machines*, vol. 19, no. 1-2, pp. 305-307, 2017. Available: 10.1007/s10710-017-9314-z.

[23] Y. Liu, S. Liu and X. Zhao, "Intrusion Detection Algorithm Based on Convolutional Neural Network", *DES tech Transactions on Engineering and Technology Research*, no., 2018. Available: 10.12783/dtetr/iceta2017/19916.

[24] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017. Available: 10.1145/3065386.

[25] D. Roy, P. Panda and K. Roy, "Tree-CNN: A hierarchical Deep Convolutional Neural Network for incremental learning", 2022.

[26] A. Mahendran and A. Vedaldi, "Visualizing Deep Convolutional Neural Networks Using Natural Pre-images", *International Journal of Computer Vision*, vol. 120, no. 3, pp. 233-255, 2016. Available: 10.1007/s11263-016-0911-8.