Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 26, S. 22-29, Temmuz 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 26, pp. 22-29, July 2021

Research Article

Network Intrusion Detection Approach Based on Convolutional Neural Network

Hakan Can Altunay[1*], Zafer Albayrak[2]

[1*] Ondokuz Mayis University, Departmant of Computer Technologies, Samsun, Turkey, (ORCID: 0000-0002-0175-239X), hakancan.altunay@omu.edu.tr
[2] Karabuk University, Faculty of Engineering, Departmant of Computer Engineering, Karabuk, Turkey, (ORCID: 0000-0001-8358-3835), zalbayrak@karabuk.edu.tr

(3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications June 11-13, 2021)

(**DOI:** 10.31590/ejosat.954966)

ATIF/REFERENCE: Altunay, H.C. & Albayrak, Z. (2021). Network Intrusion Detection Approach Based on Convolutional Neural Network. *European Journal of Science and Technology*, (26), 22-29.

Abstract

The probability of encountering cyber-attacks increases with the proliferation of internet usage and the increase in the number of network devices. Intrusion detection systems are used in order to prevent the damages caused by cyber-attacks. In this study, an intrusion detection implementation based on feature selection was performed by using a convolutional neural network in order to prevent cyber-attacks. CSE-CIC-IDS2018 dataset was used during the training and testing stages. Attributes of the dataset were trained on the preprocessing layer, classification layer, and two-layer convolutional neural network. The implementation performance was assessed through accuracy, precision, and recall metrics. A retraining stage was performed in order to resolve the over-learning problem of the network. Intrusion detection was performed through synthetic data generation within the dataset. SMOTE (Synthetic Minority Over Sampling Technique) was used for synthetic data generation. In the study, Brute Force, SQL Injection, Botnet, and DoS attacks were selected as the types of threat. Attack detection accuracy of the intrusion detection system was found 98.32% and the detection accuracy obtained after retraining was found 98.8%. Following the training performed with synthetic data added into the dataset, the neural network carried out a binary classification of the data. The performance rate of detection and classification of the data as a threat was determined as 98.7% for Brute Force, 98.5% for DoS, 98.9% for Botnet, and 99.1% for SQL Injection.

Keywords: Intrusion Detection Systems, Convolutional Neural Network, SMOTE, Deep Learning, Cyber Security.

Evrişimli Sinir Ağına Dayalı Ağ Saldırı Tespit Yaklaşımı

Öz

İnternet kullanımının yaygınlaşması ve ağa bağlı cihaz sayısının artması ile siber saldırılarla karşılaşma olasılığı artmaktadır. Siber saldırıların verdiği zararları, engellemek için saldırı tespit sistemleri kullanılmaktadır. Bu çalışmada siber saldırıların engellenmesi için, evrişimli sinir ağı kullanılarak özellik seçimine dayalı saldırı tespit uygulaması gerçekleştirilmiştir. Eğitim ve test işlemlerinde CSE-CIC-IDS2018 veri seti kullanılmıştır. Veri setindeki öznitelikler, ön işlem katmanı, sınıflandırma katmanı ve iki katmanlı evrişimli sinir ağı üzerinde eğitilmiştir. Uygulamanın performansı accuracy, precision ve recall ölçütleri ile değerlendirilmiştir. Ağın aşırı öğrenme sorununu gidermek için yeniden eğitim aşaması gerçekleştirilmiştir. Veri seti içerisinde sentetik veri üretimi gerçekleştirilerek izinsiz giriş tespiti yapılmıştır. Sentetik veri üretimi için SMOTE (Synthetic Minority Over Sampling Technique) yöntemi kullanılmıştır. Çalışmada tehdit türleri olarak Brute Force, Sql Injection, Botnet ve DoS saldırıları seçilmiştir. Saldırı tespit sistemine ait saldırı algılama doğruluğu %98.32 ve yeniden eğitim sonrası elde edilen algılama doğruluğu ise %98.8 olarak tespit edilmiştir. Veri setine eklenen sentetik veriler ile gerçekleştirilen eğitim sonunda sinir ağı, verilerin ikili sınıflandırma işlemini gerçekleştirmiştir. Verilerin tehdit olarak algılanıp sınıflandırılmasındaki başarımı, Brute Force için %98.7, DoS için %98.5, Botnet için %98.9 ve SQL Injection için %99.1 olarak bulunmuştur.

Anahtar Kelimeler: Saldırı Tespit Sistemleri, Evrişimli Sinir Ağı, SMOTE, Derin Öğrenme, Siber Güvenlik.

[*] Corresponding Author: hakancan.altunay@omu.edu.tr

1. Introduction

Intrusion detection systems are used in order to prevent cyber-attacks. These systems identify threats by monitoring data traffic on a network or system (Deng, Zhuang, & Liang, 2017; Li, Batta, & Trajkovic, 2018). Any identified threat is either conveyed to the system administrator or collected centrally by using a security event management system. Security event management systems combine data from various sources and carry out a filtering process, and, in this way, perform a true or false alarm detection (Kevric, Jukic, & Subasi, 2017).

Intrusion detection systems are categorized into two groups as signature- and anomaly-based detection systems. The method of identifying any intrusion and storing it in the database is called a signature-based detection system (Sharafaldin, Arash, & Ali, 2018). Features in the dataset and signatures in the database are crosschecked in the signature-based detection systems. If the malicious software behavior does not match with the signature in the database, the similarity test fails and the intruder accesses the system. The higher the number of signatures in the database, the longer it takes to process and analyze the huge volume of data (Alazab, Hobbs, Abawajy, & Alazab, 2014). In the anomaly-based detection system, on the other hand, the features from the dataset are trained. Data behavior is classified as normal or abnormal at the end of the training. Anomaly detection techniques are able to detect new attacks. However, high false alarm rates may occur in the changing cyber-attack mediums. Anomaly-based detection systems consist of two stages as training stage and the testing stage. Hybrid intrusion detection models are developed in order to eliminate the disadvantages of the signature- and anomaly-based detection systems. Network data and system information are combined in hybrid intrusion detection systems (Khraisat, Gondal, Vamplew, Kamruzzaman, & Alazab, 2020).

Artificial intelligence and machine learning methods are used in anomaly-based detection systems. Machine learning methods perform reinforcement, supervised and unsupervised learning to build models (Alabadi, &Albayrak, 2020). These methods are usually preferred in implementations where there is a small amount of data (Baykara, & Daş, 2019). Intrusions with unknown sources can be identified by using deep learning techniques with the increase in the amount of data. Deep learning is a machine learning method (Priyadarshini, & Barik, 2019). The main purpose of using deep learning techniques in intrusion detection systems is preventing or minimizing the attacks with signatures similar to the anomaly traffics identified on the network (Behera, Pradhan, & Dash, 2018).

In this study, it was aimed to identify and classify the attacks on the computer networks by using a convolutional neural network. CSE-CIC-IDS 2018 dataset was used in our study. Dataset has 82 features apart from label information. In the data preprocessing stage, ICFlowMeterV3 and attributes of source IP, target IP, source port, flow identity, and intrusion time that would not be used in the threat model were removed from the dataset. In addition, attributes of Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk, Fwd Avg Bytes/Bulk, Bwd PSH Flags, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate and Bwd Avg Bulk Rate with null values within the dataset were also removed. Data received by the networks were classified through the recommended method and attack detection accuracy was determined. Synthetic data generation was performed through SMOTE. It was aimed to increase the performance rate in intrusion detection and classification by retraining the entire neural network through synthetic data. Values obtained at the end of retraining signify that the model properly identified and classified the new intrusion data.

Accuracy, precision, and recall values obtained towards the attack detection at the end of the recommended method were presented. In the comparison of these values and the results of other implementations in the literature where deep learning techniques were used, higher performance was reached in attack detection. It was determined that the network performs over-learning in types of attacks with a small amount of data. The over-learning problem was prevented by increasing the number of types of attacks that were low in number within the dataset through synthetic data generation. The recommended method is different from other studies in the literature in detecting the Botnet, DoS, SQL Injection, and Brute Force attacks by using a convolutional neural network and synthetic data generation.

A literature review was performed in the second section of the study. In the third section, a threat model was created and data preprocessing, classification, and convolutional neural network layers included in the intrusion detection approach were explained. In the fourth section, the results belong to the recommended approach were analyzed. In the last section, the results obtained were evaluated and studies projected to be carried out in the future were addressed.

2. Related Works

Several machine learning methods such as support vector machines, k-nearest neighbors, decision tree and artificial neural networks, have been used in the development of intrusion detection systems. Deep learning approaches such as long short-term memory, convolutional neural network, and recurrent neural network have also been preferred with the increase in the amount of data in recent years. These studies were explained in accordance with the method used and results obtained.

In 2019, Botnet attacks within the CSE-CIC-IDS2018 dataset were detected by Ring et al. by using multi-layer detectors. In addition, optimization was performed on the model displaying extreme consistency with the default hyperparameters by using the GridSearch method. Botnet attacks were classified with a 99.97% accuracy rate (Ring, Wunderlich, Scheuring, Landes, & Hotho, 2019). In another study conducted by Kanimozhi et al., the performance of the recommended model in the classification of the types of attacks was analyzed by using the CSE-CIC-IDS2018 dataset. Intrusions on the dataset were detected through six different machine learning methods (Decision Tree, Gaussian Naive Bayes, MLP, Random Forest, KNN, QDA) and 10-fold cross-validation. In the study, the decision tree method performed the highest intrusion detection with a 96% accuracy rate (Kanimozhi, & PremJacob, 2019). In another study conducted by Zhou and Pezaro designing an intrusion detection system on the CSE-CIC-IDS2018 dataset through long short-term memory (LSTM), a data addition procedure was implemented through SMOTE for the numbers of types of attack within the dataset to be close to each other. The intrusion detection performance of the recommended model on

the dataset that was expanded with the added data was analyzed. Precision, recall, F-score, and accuracy rate values were provided in order to evaluate the performance of the intrusion detection system. It was revealed that low performance was obtained in detecting the intrusions in the datasets where the numbers of types of attack were not close to each other (Zhou, & Pezaro, 2019). In a Blockchain-based service for drone-delivered services, an intrusion detection system was developed by using the CSE-CIC-IDS2018 dataset. Convolutional neural network (CNN), recurrent neural network (RNN), support vector machine (SVM) and random forest methods were used in order to identify the attack packages within the dataset. It was revealed that the highest performance rate was achieved with 98.75% in the detection of Botnet attacks (Yin, Zhu, Fei, & He, 2017). In 2019, denial-of-service (DoS) attacks were detected on the CSE-CIC-IDS2018, CICIDS2017, and CICDOS training sets through the random forest method. As a result of the study conducted, 0.999, 0.992, and 0.995 F-score values were obtained respectively for the datasets (Ferrag, & Maglaras, 2019). In another study carried out in 2019, it was determined that the highest accuracy rate was reached by the decision tree algorithm among the algorithms ofrandom forest, deep neural networks, decision tree and k-nearest neighbors used in the collective learning model (Filho, Frederico, Silveira, Junior, & Silveira, 2019). Abdulhammed et al. created an intrusion detection system that works on datasets including various numbers of types of attacks by using the CSE-CIC-IDS2018 dataset. A 99.99% accuracy rate was achieved in the implementation where there is limited change in the amount of data (Abdulhammed, Faezipour, Abuzneid, & Abumallouh, 2018). Another intrusion detection system was designed by Odabas and Pehlivanoglu on the CSE-CIC-IDS2018 dataset through a two-level hybrid machine learning method. The intrusion detection was performed by using CNN, Random Forest, Light Gradient Boosting Machine (LGBM), CNN + Random Forest, LGBM + Random Forest, and Random Forest + Random Forest machine learning methods. It was revealed that the best result was achieved through CNN + Random Forest method with a 98% accuracy rate and 0.86 F-score value (Atay, Odabas, & Pehlivanoglu, 2019). In another implementation performed by using artificial neural networks, an anomaly-based intrusion detection system was designed on the CSE-CIC-IDS2018 dataset. The detection accuracy of the recommended system was revealed at the end of the study according to the types of attacks within the dataset (Karaman, Turan, & Aydın, 2020). In the hybrid model developed by Sun et al. by using convolution neural network and long short-term memory (LSTM), category weight optimization method was used. In the model where the temporal and spatial features of network traffic were extracted by using CNN + LSTM hybrid network, the overall accuracy rate was determined as 98.67% in the multiple classification (Sun, Liu, Li, Lu, Hao, & Chen, 2020). In another study conducted for intrusion detection, a new algorithm based on feature selection was recommended. The algorithm was performed on various datasets. It was revealed through the algorithm that the recommended cosine similarity method has a faster convergence than the sigmoid similarity method (Alazzam, Sharieh, & Sabri, 2020). An intrusion detection system based on a deep learning model optimized by using rule-based hybrid feature selection was developed by Femi et al. It was expressed that the number of selected features would not affect the detection accuracy of the feature selection algorithms in that model. It was revealed in the study that the number of selected features was directly proportional to the performance of the base classifier (Femi, Sakinat, Adebay, Adebola, & Joseph, 2020). In another intrusion detection system based on a deep learning model developed by feature selection, DoS attacks were detected by using two different datasets as CSE-CIC-IDS2018 and KDD CUP99. The model was developed by using a convolutional neural network and its performance was evaluated through comparison with a recurrent neural network (RNN) (Jiyeon, Jiwan, Hyunjung, Minsun, & Eunjung, 2020). In another study designed by Einy et al., the artificial neural network and IDS/IPS were used together for the manual detection of malicious traffic in the network. A metaheuristic method was used for feature selection and fuzzy logic was used for anomaly detection. The network intrusion was prevented at the rate of 96.11% at the end of the study (Einy, Oz, & Navaei, 2021).

The implementations performed by using machine learning and deep learning techniques on the CSE-CIC-IDS2018 dataset in the literature were explained above. The algorithms used in these implementations performed for detection and classification of the intrusions were compared and inferences were made for future studies. The studies conducted by using a convolutional neural network on the CSE-CIC-IDS2018 dataset and the accuracy values obtained are presented in Table 1.

Table 1. Comparison of the Intrusion Detection Systems Using CNN

Reference	Method	Type of Attack	Results
(Yin et al.)	SVM, RNN, CNN	DoS, User to Root, Probe, Root to Local	98.75%
(Atay, Odabas, and Pehlivanoglu)	CNN, CNN + Random Forest	All	98%
(Sun et al.)	CNN + LSTM	All	98.67%
(Jiyeon et al.)	CNN, RNN	DoS, DDoS	95%

3. Matherial and Method

Today, intrusions are carried out in various ways such as interruption of network services, alteration of data, and sniffing. Intrusion detection systems are used to identify and minimize these intrusions. These systems are designed through threat models created according to the types of attacks within the current dataset (Tuptuk, & Hailes, 2018). The aim of this study is to analyze the random package behaviors reaching a computer network from a dataset through a convolutional neural network (CNN). It was determined whether the packages reaching the intrusion detection system were threats or not, then they were classified upon the determination of the threat model they belong to. The neural network was subjected to training and testing processes for the types of attacks such as DoS, Brute Force, SQL Injection, and Botnet.

Table 2. Confusion Matrix

		Predicted	
Actual		Negative	Positive

	Negative	TN_i	FP_i
	Positive	FN_i	TP_i

$$Accuracy = \frac{TPi + TNi}{TPi + TNi + FPi + FNi} \quad (1)$$

$$Precision = \sum_{i=1}^{N} \frac{TPi}{TPi + FPi} \quad (2)$$

$$Recall = \sum_{i=1}^{N} \frac{TPi}{TPi + FNi} \quad (3)$$

3.1. Threat Modeling

In the model created, an intruder may threaten the network server or access other devices in the network by using the server. In the cyber-attacks on computer networks, computers are tried to be disabled through malicious software in addition to passive sniffing. The malicious software tries to access other devices over the server without creating an abnormality in the network traffic. In the study, the new data packages were also created through SMOTE during a cyber-attack since it was aimed for the recommended approach to identify and classify all the malicious network status generated.

3.2. System Design

The recommended design architecture is presented in Figure 1. Our input data consists of the attack packages within the CSE-CIC-IDS2018 dataset. In the preprocessing stage in the architecture, the dataset consists of 82 attributes. The packages following the data preprocessing are subjected to the binary classification and it is determined whether they are threats or not (Akashdee, Manzoor, & Kumar, 2017). Then, it is determined what types of attacks the packages, which are characterized as attacks, generate. The detection process basically analyses the network behavior patterns through attack packages. The recommended intrusion detection system flow chart is presented in Figure 1. Detection modules are defined for abnormal situations. The detection algorithm affects each detection module when the output is obtained. Detection modules with abnormalities are analyzed for intrusion detection and intruders are identified. It is determined whether the packages reaching the intrusion detection system are threats or not, then they are classified upon the determination of the threat model they belong to.

All data must be between 0 and 1 before the testing stage (Chandra, Khatri, & Simon, 2019). The normalization process is implemented to the data that are not between 0 and 1 by using a k-means clustering algorithm. Different detection modules are used for feature learning. Feature vectors of the packages within each detection module are processed through a convolutional neural network (CNN). Learned features are classified in order to create output at the end of the detection process. An output layer fully connected to the CNN was added for the classification process. The algorithm labels the data package upon threat analysis when it encounters an attack. Brute force attacks were labeled as “1”, DoS attacks as “2”, Botnet attacks as “3”, and SQL Injection attacks as “4”. A new layer is added to the classification process as the number of attacks increases and the entire neural network is retrained. There are 286.191 Botnet, 513 Brute Force, 53 SQL Injection, and 1.289.544 DoS attack packages in the CSE-CIC-IDS2018 dataset, as presented in Table 3.

Table 3. Types and Numbers of Attacks Within The Dataset

Label	Number
Botnet	286.191
SQL Injection	53
Brute Force	513
DoS	1.289.544
Infiltration	93.063
Benign	2.865.035
Total	4.525.399

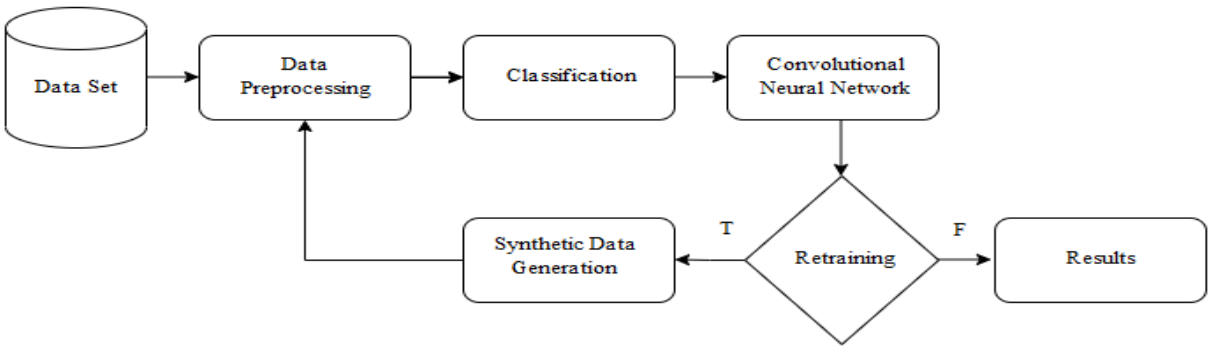

Figure 1. Recommended Intrusion Detection System Flow Chart

3.3. Data Generation Through SMOTE

The numbers of types of attack in the current dataset are different from each other. Therefore, there is irregularity among the attack packages within the dataset. Data was sampled in the preprocessing stage and the imbalanced distribution within the dataset was eliminated. The sampling process was performed through the synthetic data generation model named SMOTE (Synthetic Minority Over Sampling Technique). SMOTE generates new data by using K-Nearest Neighbors (KNN) algorithm. In the SMOTE, the difference between the feature vector and the nearest neighbor of the feature vector is multiplied with a random number between 0 and 1. New synthetic data are generated by adding the result obtained to the feature vector (Yavas, Guran, Uysal, Manzoor, & Kumar, 2020). The algorithm steps of the data generation through SMOTE are itemized below and the operation logic is presented in Figure 2.

Step 1: The k-nearest neighbor of each sample that belongs to the class with a small amount of data is found.

Step 2: The difference between the class with the small amount of data and its k-nearest neighbor is determined.

Step 3: A random number (x) between 0 and 1 is multiplied with the difference value found in Step 2.

Step 4: New data generation is performed according to the following equation:

$$a_{new} = a_0 + (a_1 - a_0) * x \quad (4)$$

Step 5: The first four steps are repeated for each new data to be generated.

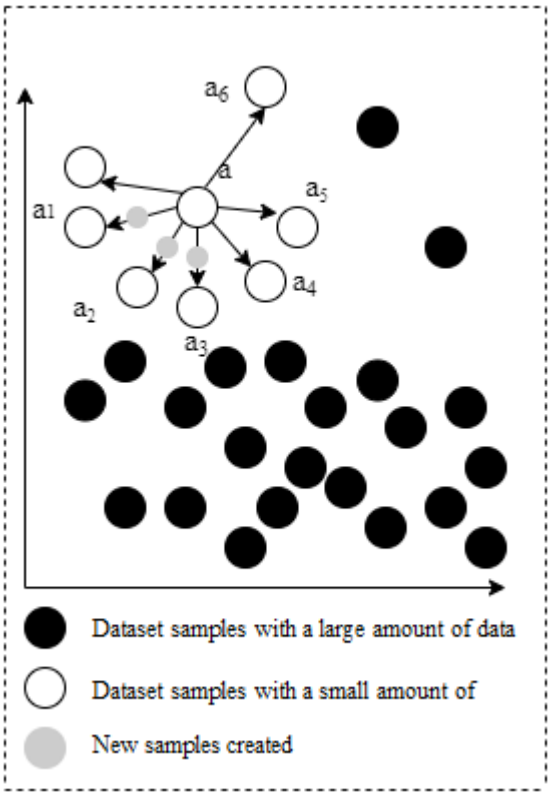

Figure 2. Data Generation Through SMOTE

If there are one or more attacks that have not been observed before in a detection module, the output neurons in the stage of classification have values close to each other. In this case, it becomes difficult to identify the network attacks on the computer systems. The performance rate of the intrusion detection system is wanted to be increased through retraining with new data that the system has not encountered before. The classification stage outputs must be between 0 and 1 for all classes in the testing stage. Each output that is not between 0 and 1 will be labeled as "unknown" and it will be regarded as a sample that was not encountered before. These labeled samples are sent to the current training set. All neural network is retrained with these attack classes that were not encountered before. The amount of synthetic data generated through SMOTE is presented in Table 4. The sample data to be generated were implemented for the data including 5% or fewer samples in our dataset. Therefore, the numbers of Brute Force, SQL Injection, and Infiltration attack packages were updated.

Table 4. Amount of Data Generated Within The Dataset

Label	Number
Botnet	286.191
SQL Injection	286.191
Brute Force	286.191
DoS	1.289.544
Infiltration	286.191
Benign	2.865.035
Total	5.299.343

3.4. Convolutional Neural Network and Feature Learning

As presented in Table 5, two one-dimensional convolutional layers, pooling layer, flatten layer, and ReLU activation function were used in the convolutional neural network. In addition, an output layer including two fully connected layers with 256 neurons with 0.2 dropout value and softmax activation function with 15 neurons was used. Epoch was determined as 100 and Batchsize was determined as 1000. 80% of the attack numbers to be used within the dataset were used for training (1.261.040) and 20% were used for testing (315.260). In the dataset, the label of each sample is determined according to the type of network attack within the related detection module. If the sample does not include any attack packages, this is called a "normal situation." Packages share the same label in a normal situation. In each convolutional layer, the input feature maps are combined with the convolutional core learned during the training process.

Table 5. CNN Structure Parameters and Values

Parameter	Value
Convolution	2
Pooling	2
Flatten	1
Dropout	0.2
Dense_1	Activation=ReLU
Dense_2	Activation =ReLU
Dense_3	Activation =Softmax

Then, the activation function is implemented in order to create the feature map of the next layer. In the pooling layer, the size of the input is decreased by summarized on the local values of the feature maps received from the previous layer. The convolutional neural network (CNN) presented in Figure 3 is

created by using the convolution and pooling processes defined above.

CNN layers are divided into 4 groups. The first and second layers create the convolutional input and feature maps. Then, the rectified linear unit (ReLU) layer processes the output of the previous layer. Pooling is performed in order to select the maximum feature map. Finally, the normalization process is implemented on the values from various feature maps received from the previous layer. A ReLU and normalization process are performed in the third layer, which is a convolutional layer. In the fourth layer, feature maps received from the previous layer are combined. In this stage, a fully connected output layer is added to CNN in order to match it with the output classes of the learned features.

The ReLU is the activation function used on all samples. The network performs the learning process in a non-linear way. It is preferred in multilayer neural networks since its calculation load is low.

$$ReLU = g(z) = max(0, z) \quad (5)$$

The Softmax, on the other hand, is an activation function that is particularly preferred in the output layer of the deep learning models when more than two classifications are required. It generates a random value between 0 and 1 and enables the determination of the probability of data regarding its inclusion in a class (Yang, Cheng, & Chuah, 2019).

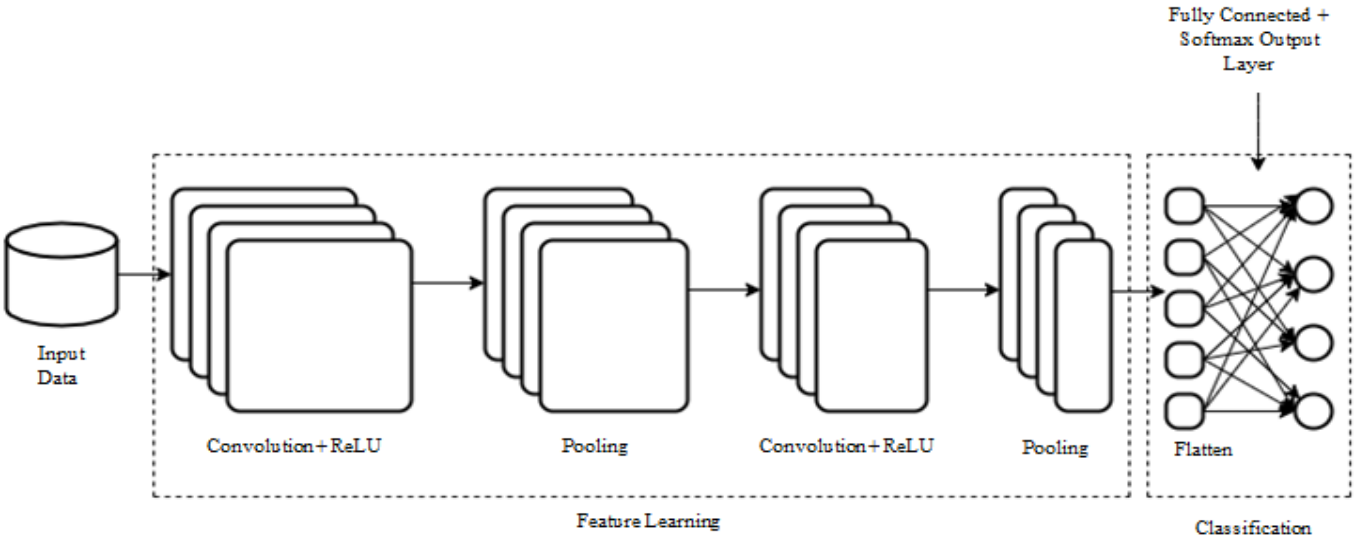

Figure 3: CNN Architecture Used in The Study

4. Research Results and Discussion

The accuracy, precision, and recall values were obtained as the performance results of the intrusion detection implementation following the training of our algorithm on the dataset. The F-score performance metric was not analyzed since the precision and recall values were found close to each other. It is observed that the algorithm identified all the samples of the attack classes encountered during the entire training process with high accuracy. The overall detection accuracy is 98.32%. This result signifies that there is a low rate of false positives generated and a small number of samples classified as the normal process. The entire neural network was retrained by performing synthetic data generation within the dataset. In the study, a total of 571.816 samples were labeled for retraining. The overall detection accuracy of the approach following the retraining was determined as 98.8%. The small number of labeled samples causes the accuracy of the model to decrease or the samples of the other classes to be labeled by mistake. The misclassification of the attacks that were not encountered before prevents the intrusion detection model from identifying the new attack types occurring. The performance metrics of the approach are presented in Table 6.

Table 6. Performance Metrics of The Network with The Data Within The Dataset

Performance Metrics	Results of The Attacks Performed with The Data Within The Dataset	Results of The Attacks Performed with The Addition of The Data Generated
Precision	99.6%	99.5%
Recall	98.3%	98.1%
Accuracy	98.32%	98.8%

The performance rate of the samples of types of attack within the dataset in terms of classification accuracy during the entire training process is presented in Figure 4. The recommended approach detected and classified the Brute Force, DoS, and Botnet attack types with high accuracy. The SQL Injection attack samples with 100% classification accuracy, on the other hand, signify that our network has an over-learning problem in the types of attacks with a small amount of data. In order to eliminate the over-learning problem, synthetic data generation was performed and the entire neural network was retrained with the types of attacks that were not encountered before. High accuracy was obtained in various attack classes

when the model was retrained with attack classes that were not encountered before. This result signifies that the retraining process step adapts itself to the neural network. The performance rate was found 99.1% in the SQL Injection threat type after the data was added, while this rate was 100% before the data was added. Therefore, the implemented approach cannot reach accurate results with a small amount of data. The numbers of correctly or wrongly classified attack packages on the synthetic data-added dataset in the implementation are presented in Table 7.

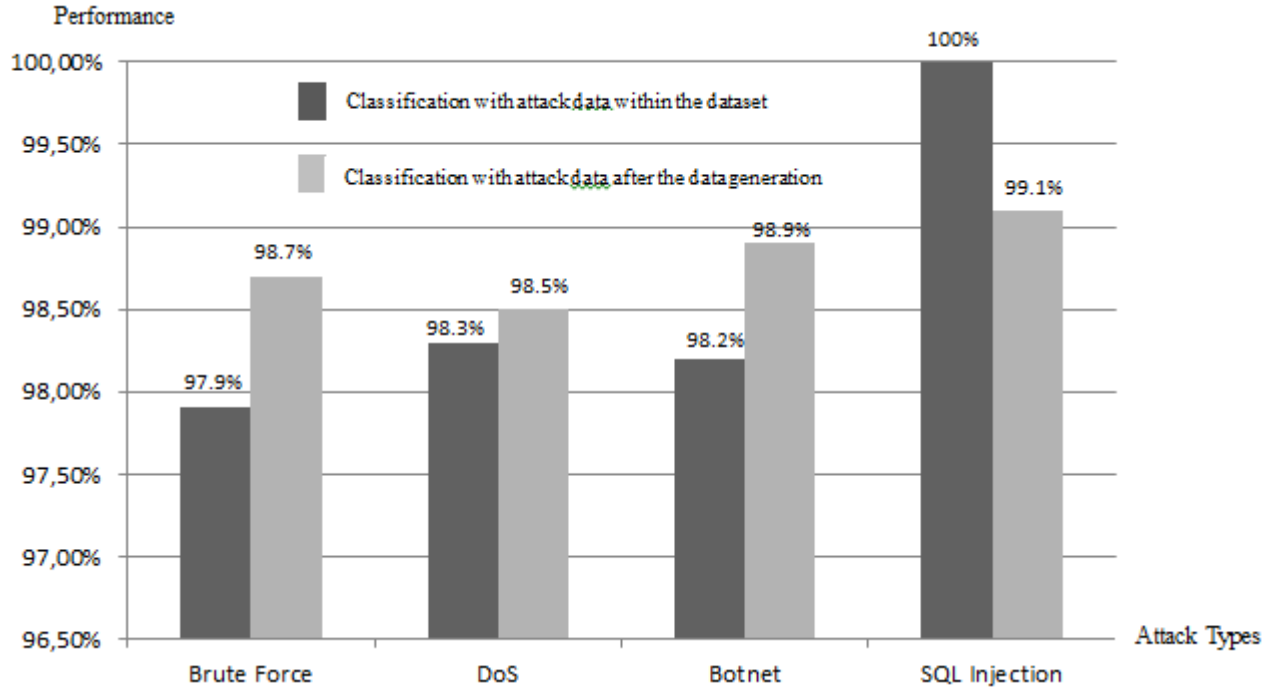

Figure 4: Classification Performance of The Recommended Method

Table 7. Numbers of Correctly or Wrongly Classified Attack Packages

Attack Type	Correctly Classified Attack Packages	Wrongly Classified Attack Packages
Botnet	283.042	3.149
DoS	1.270.200	19344
Brute Force	282.470	3671
SQL Injection	283.615	2576

4. Conclusion and Future Studies

In this study, the intrusion detection systems using deep learning were analyzed and an intrusion detection implementation was performed by using a convolutional neural network (CNN) based on feature extraction. Modeling and feature learning of the temporary network behavior patterns of the server on the network or the devices connected to that server were performed through the convolutional neural network (CNN) method. In the study conducted through this method, Botnet, SQL Injection, Brute Force, and DoS attack types were identified and classified. It was determined that the network had an over-learning problem in the SQL Injection attack type with a small amount of data. Higher accuracy was obtained in terms of detection and classification processes of the attack types when the entire network was retrained with the new data generated through SMOTE compared to the data in the current dataset. In the study, a 98.8% accuracy rate was obtained with hyperparameters, selected features, and synthetic data generation. Higher performance was reached in terms of accuracy, precision, and recall performance metrics compared to the CNN-based intrusion detection models performed in the literature. In the future, intrusion detection systems can be designed through hybrid- and rule-based methods by using various datasets. Comparisons can be made between intrusion detection and classification values of the designed models and intrusion detection method using a convolutional neural network.

References

Deng, R., Zhuang, P., & Liang, H. (2017). CCPA: Coordinated Cyber-Physical Attacks and Countermeasures in Smart Grid. IEEE Transactions on Smart Grid, 2420–2430.

Li, Z., Batta, P., & Trajkovic, L. (2018). Comparison of machine learning algorithms for detection of network intrusions. In

2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 4248–4253.

Kevric, J., Jukic, S., & Subasi, A. (2017). An effective combining classifier approach using tree algorithms for network intrusion detection. Neural Computing and Applications, 1051–1058.

Sharafaldin, I., Arash, H. L., & Ali, A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. 4th International Conference on Information Systems Security and Privacy (ICISSP). Portekiz.

Alazab, A., Hobbs, M., Abawajy, J., & Alazab, M. (2014). Using response action with intelligent intrusion detection and prevention system against web application malware. Information Management and Computer Security.

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2020). Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine. Electronics.

Alabadi, M., & Albayrak, Z. (2020). Q Learning for Securing Cyber-Physical Systems: A survey. (2020). International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 1-13.

Baykara, M., & Daş, R. (2019). Saldırı Tespit Ve Engelleme Araçlarının İncelenmesi. Dümf Mühendislik Dergisi, 57-75.

Priyadarshini, R., & Barik, R.K. (2019). A Deep Learning Based Intelligent Framework to Mitigate DDoS Attack in Fog Environment. Journal of King Saud University - Computer and Information Sciences.

Behera, S., Pradhan, A., & Dash, R. (2018). Deep neural network architecture for anomaly based intrusion detection system. In 5th International conference on Signal Processing and Integrated Networks, 270-274.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho A. (2019). A Survey of Network-based Intrusion Detection Data Sets. Cryptography and Security.

Kanimozhi, V., & PremJacob, T. (2019). Artificial Intelligence based Network Intrusion Detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. ICT, 211-214.

Zhou, Q., & Pezaro, D., (2019). Evaluation of machine learning classifiers for zero-day intrusion detection, an analysis on CIC-AWS-2018 dataset. arXiv abs/190.03685v1.

Yin, C., Zhu, Y., Fei, J., He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE
Access, 21954-21961.

Ferrag, M.A., & Maglaras, L. (2019). Deliverycoin: An ids blockchain-based framework for drone-delivered services. Computers, 58.

Filho, F., Frederico, A., Silveira, F., Junior, A., & Silveira, G. (2019). Smart detection: An online approach for DoS/DDoS Attack detection using machine learning. Security and Communication Networks.

Lin, P., Ye, K., & Xu, C.Z. (2019). Dynamic Network Anomaly Detection System by Using Deep Learning Techniques. In: Da Silva, D., Wang, Q., Zhang, LJ. (eds) Cloud Computing – CLOUD 2019. CLOUD 2019. Lecture Notes in Computer Science, vol 11513. Springer, Cham.

Abdulhammed, R., Faezipour, M., Abuzneid, A., & Abumallouh, A. (2018). Deep and machine learning approaches for anomaly-
based intrusion detection of imbalanced network traffic. IEEE Sensors Letters, 1-4.

Atay, R., Odabaş, D. E., & Pehlivanoğlu, M. K. (2019). İki Seviyeli Hibrit Makine Öğrenmesi Yöntemi İle Saldırı Tespiti. Dergipark, 258-272.

Karaman, M., Turan, M., & Aydın M. A. (2020). Yapay sinir ağları kullanılarak anomali tabanlı saldırı tespit modeli uygulaması. European Journal of Science and Technology Special Issue, 17-25.

Sun, P., Liu, P., Li, Q., Lu, X., Hao, R., & Chen, J. (2020). DL-IDS: extracting features using CNN-LSTM hybrid network for
intrusion detection system. Security and Communication Networks.

Alazzam, H., Sharieh, A., & Sabri, K. E. (2020). A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Security and Communication Networks Expert Systems with Applications, 148, 113249.

Femi, E. A., Sakinat, O. F., Adebayo, A. A., Adebola, O. A., & Joseph, B. A. (2020). Network intrusion detection based on deep
learning model optimized with rule-based hybrid feature selection. Information Security Journal.

Jiyeon, K., Jiwan, K., Hyunjung, K., Minsun, S., & Eunjung, C. (2020). CNN-Based network intrusion detection against denial-of-service attacks. Electronics.

Einy, S., Öz, C., & Navaei, N. V. (2021). The Anomaly- and Signature-Based IDS for Network Security Using Hybrid Inference Systems. Mathematical Problems in Engineering.

Tuptuk, N., & Hailes, S. (2018). Security of smart manufacturing systems. ELSEVIER.

Karataş, G., Demir, Ö., & Şahingöz, Ö. K. (2019). A deep learning based intrusion detection system on GPU's. International Conference 11th Edition Electronics computer and Artificial Intelligence.

Akashdee, P., Manzoor, I., & Kumar, N. (2017). A feature reduced intrusion detection system using ANN classifier. ELSEVİER.

Chandra, A., Khatri, S., & Simon, R. (2019). Filter-based attribute selection approach for intrusion detection usin k-means clustering and sequential minimal optimization technique. Amity International conference on Artificial Intelligence, 740-745.

Yavaş, M., Güran, A., Uysal, M., Manzoor, l., & Kumar, N. (2020). Covid 19 veri kümesinin SMOTE tabanlı örnekleme yöntemi uygulanarak sınıflandırılması. European Journal of Science and Technology.

Yang, H., Cheng, L., Chuah, M. C. (2019). Deep learning based network intrusion detection for SCADA systems. IEEE Conference on Communications and Network Security: Workshops: CPS: International Workshop On Cyber-Physical Systems Security.