Mühendislik Bilimleri ve Tasarım Dergisi
12(3), 517-530, 2024
e-ISSN: 1308-6693

Journal of Engineering Sciences and Design
DOI: 10.21923/jesd.1417622

Araştırma Makalesi Research Article

DEEP LEARNING BASED NETWORK INTRUSION DETECTION

Güneş HARMAN[1*], Emine CENGİZ[1]

[1] Department of Computer Engineering, Yalova University, Yalova, Turkey

Keywords

Deep Learning, Convolutional Neural Network, Intrusion Detection, NF-BoT-IoT, Cyber Security.

Abstract

As a direct consequence of the unrelenting march of technological innovation, the use of the Internet has become an unavoidable condition for the life of modern humans. The Internet has increased both the quantity and range of situations in which information products can be useful or non-useful. It's no surprise that as the number of different systems and users has grown, so have the number of different ways to exploit those systems. A security issue has arisen with such diversity and growth. Its diversity and increase in quantity introduce new system weaknesses and thus new attack strategies. Methods for detecting both internal and external attacks are suggested as a solution to this issue. The purpose of this research, a Convolutional Neural Network was utilized to identify intrusions, also known as attacks for the imbalanced class distribution in the NF-BoT-IoT data set, Synthetic Minority Over Sampling Technique, Random Over Sampling and Random Under Sampling methods were used. K-Fold Cross Validation, one of the strategies for splitting the data set, was utilized to evaluate the performance of classification models and to train the developed model. The model's performance was evaluated using the accuracy, precision, recall, and F1-score performance criteria.

DERİN ÖĞRENME TABANLI AĞ SALDIRI TESPİTİ

Anahtar Kelimeler

Derin Öğrenme, Evrişimsel Sinir Ağları, Saldırı Tespiti, NF-BoT-IoT, Siber Güvenlik.

Öz

Teknolojik yeniliklerin amansız ilerleyişinin doğrudan bir sonucu olarak, İnternet kullanımı modern insanın yaşamı için kaçınılmaz bir koşul haline gelmiştir. İnternet, bilgi ürünlerinin yararlı ya da yararsız olabileceği durumların hem miktarını hem de çeşitliliğini artırmıştır. Farklı sistemlerin ve kullanıcıların sayısı arttıkça, bu sistemleri istismar etmenin farklı yollarının sayısının da artması şaşırtıcı değildir. Bu çeşitlilik ve büyümeyle birlikte bir güvenlik sorunu ortaya çıkmıştır. Çeşitlilik ve miktar artışı yeni sistem zayıflıklarını ve dolayısıyla yeni saldırı stratejilerini beraberinde getirmektedir. Bu soruna çözüm olarak hem iç hem de dış saldırıları tespit etmek için yöntemler önerilmektedir. Bu araştırmanın amacı, NF-BoT-IoT veri setindeki dengesiz sınıf dağılımına yönelik saldırı olarak da bilinen izinsiz girişleri tespit etmek için bir Evrişimsel Sinir Ağı kullanılmış, Sentetik Azınlık Örnekleme Tekniği, Rastgele Aşırı Örnekleme ve Rastgele Alt Örnekleme yöntemleri kullanılmıştır. Sınıflandırma modellerinin performansını değerlendirmek ve geliştirilen modeli eğitmek için veri setini bölme stratejilerinden biri olan K-Fold Cross Validation kullanılmıştır. Modelin performansı doğruluk, kesinlik, duyarlılık ve F1-skor performans kriterleri kullanılarak değerlendirilmiştir.

Alıntı / Cite

Harman, G., Cengiz, E., (2024). Deep Learning Based Network Intrusion Detection, Journal of Engineering Sciences and Design, 12(3), 517-530.

Yazar Kimliği / Author ID (ORCID Number)
G. Harman, 0000-0001-5413-124X
E. Cengiz, 0000-0002-6695-9500

Makale Süreci / Article Process	
Başvuru Tarihi / Submission Date	11.01.2024
Revizyon Tarihi / Revision Date	25.07.2024
Kabul Tarihi / Accepted Date	06.08.2024
Yayım Tarihi / Published Date	26.09.2024

[*] İlgili yazar / Corresponding author: gunes.guclu@yalova.edu.tr, +90-226-815-5336

DEEP LEARNING BASED NETWORK INTRUSION DETECTION

Güneş HARMAN[1†], Emine CENGİZ[1]
[1] Department of Computer Engineering, Yalova University, Yalova, Turkey

Highlights

- Imbalanced Dataset Handling
- Investigating the influence of resampling methods and varying K values on model.
- Network threats using deep learning (DL) techniques, specifically focusing on the NF-BoT-IoT dataset.

Purpose and Scope

Due to relentless technological innovation, the Internet has become an indispensable aspect of modern human life, expanding the scope of situations where information products are either beneficial or not. The growing number of systems and users has led to increased vulnerabilities and various ways to exploit them. This diversity poses security challenges, prompting the need for methods to detect internal and external attacks. The purpose of this study is to explore the use of Convolutional Neural Network (CNN), a deep learning (DL) method, for suggesting Intrusion Detection Systems (IDS). The study specifically focuses on utilizing the NF-BoT-IoT dataset. The study aims to contribute to the field of intrusion detection by investigating the use of CNN with a specific focus on the NF-BoT-IoT dataset. The exploration of re-sampling techniques and K-Fold cross validation provides insights into the robustness and generalization capabilities of the proposed IDS approach.

Design/methodology/approach

The study employs the NF-BoT-IoT dataset, suggesting a focus on IoT related network traffic and potential botnet activities. Before using the dataset for training a CNN model, the data undergoes preprocessing. This includes tasks such as data cleaning, normalization, and re-sampling. Various re-sampling techniques are applied to address potential imbalances in the dataset. The techniques mentioned include Synthetic Minority Over Sampling Technique (SMOTE), Random Over Sampling, and Random Under Sampling. These techniques aim to handle imbalanced class distributions, which is common in intrusion detection datasets. The CNN model is trained using the preprocessed and resampled data. K-Fold cross validation is employed for model training. K-Fold cross validation involves dividing the dataset into K subsets and using each subset as a testing set while the K-1 remaining subsets are used for training. The study explores different values of K (5, 7, and 10) in this process. The study assesses the performance of the CNN model considering the influence of both resampling techniques and the choice of K value in the K-Fold cross validation. Performance metrics like accuracy, precision, recall, and F1 score might be used to evaluate the effectiveness of the proposed approach.

Findings

In this study, the impact of resampling methods and different K values on model performance was examined. The findings revealed a consistent 77% accuracy across three distinct K values for both SMOTE and Random Over Sampling methods. For Random Under Sampling, a 77% accuracy was achieved for K values of 5 and 7, while a 76% accuracy was observed for K value 10.
Various metrics, including accuracy, precision, recall, and F1-score, were employed to evaluate the developed models, and it was noted that these metrics yielded identical results. A comparison was made with a prior study on the NF-BoT-IoT dataset (Cengiz and Harman, 2022), where Random Forest (RF), K-Nearest Neighbor (KNN), Support Vector Machines (SVM), and Artificial Neural Network (ANN) were used for binary classification. In that research, RF achieved 99.4% accuracy, KNN achieved 82.7%, SVM achieved 96.7%, and ANN achieved 60.7%. Notably, in this current investigation, the CNN model outperformed ANN, attaining a higher accuracy of 77%.

Originality

This study is important in terms of evaluating the results obtained by using deep learning and applying different resampling techniques on a dataset with unbalanced class distribution.

† Corresponding author: gunes.guclu@yalova.edu.tr, +90-226-815-5336

1. Introduction

The Internet is a communication network that enables the sending and receiving of data and information via computers and other intelligent devices using a suitable internet protocol (TCP/IP) (Sarkar et al., 2015). Due to the pandemic (Covid19) that began in 2020, internet usage is highly popular and continues to grow. The usage of social networking sites, online meetings, and several applications has become an everyday occurrence. As financial issues, such as money transfers and electronic commerce are added it has become imperative to implement security measures in all networks that receive services over the internet network, give services over the internet network, or contact the internet network.

The widespread adoption of internet technology has simplified many aspects of people's life. Nonetheless, this circumstance has resulted in major risks over time. There are developments in security technologies with the increase in the number and types of security related threats (Mandal and Kösesoy, 2023). Its primary target is software such as authentication and access control, which prevents an unauthorized person from getting or altering their access information by either preventing access to the information altogether or by limiting who may receive it.

By developing firewalls and anti virus software, material and moral damages are drastically avoided. But, when threats increase, this software is insufficient. Intrusion Detection Systems (IDS) are real time software tools used to safeguard device communication and detect network intrusions (Vishwakarma and Kesswani, 2022). IDS enhances the environment's security by monitoring network activity and analyzing network traffic for the detection of attacks and threats. IDSs are divided into two categories: Anomaly based and signature based. Anomaly based IDS attempts to identify normal and abnormal activity on the data and alerts network management (Butun et al., 2013). Signature based IDSs, on the other hand, attempt to detect the attack by comparing the information collected from the incoming connection to the signature database (Otoum and Nayak, 2021).

Machine Learning (ML) and Artificial Intelligence (AI) are used a lot in IDS (Altunay and Albayrak, 2021). The detection accuracy of IDS has been greatly enhanced by these approaches (Mijalkovic and Spognardi, 2022). Yet, it has several drawbacks and restrictions. The processing of data, in particular, needs human interaction and specialized expertise (Shone et al., 2018). Another unfavorable feature is that as network complexity rises, learning activities decline. ML methods in IDS are typically used for applications with insufficient data (Dina and Manivannan, 2021). With the rise in data volume, techniques like Deep Learning (DL) have been employed to identify attacks. DL is a sub branch of ML (Priyadarshini and Barik, 2022). The primary objective of employing DL techniques in IDS is to detect, prevent or mitigate network defined attacks (Behera et al., 2022).

The main contributions of this study are summarized as follows:

1. In this study, we propose a Convolutional Neural Network (CNN) model using the NF-BoT-IoT dataset and explore how deep learning methods can be used to detect network attacks.

2. To address data imbalance, we employed various resampling techniques (SMOTE, Random Over Sampling, Random Under Sampling) and evaluated their impact on the performance of the CNN model. These techniques contribute to a more effective management of imbalanced class distributions.

3. We tested the generalization capability of the model by using the K-fold cross-validation method. We examined the impact of different K values (5, 7, and 10) on model performance and evaluated the role of this method in improving model accuracy.

4. In this study, we compared the performance of the CNN model with other machine learning models previously applied to the NF-BoT-IoT dataset, and demonstrated that the CNN performed better under certain conditions. This emphasizes the effectiveness and advantages of the CNN model.

2. Literature Review

In the past few years, many studies have been done on IDSs using DL algorithms. These investigations are intended to detect potential network or system attacks.

Altunay and Albayrak (2021) used CNN to make an attack detection application based on feature selection to prevent cyber-attacks in their studies. The CSE-CIC- IDS2018 dataset was utilized in their research. Using SMOTE approach to detect intrusions by generating synthetic data. The categorization success rates were 98.7% for Brute Force, 98.5% for DoS, 98.9% for Botnet, and 99.0% for SQL Injection as determined by the research.

Idrissi et al. (2021) employed DL techniques to identify network attacks in IoT systems using the Bot-IoT dataset and compared them. They utilized CNN, Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) as DL techniques. Using CNN, they attained the greatest success rate (99.94%) for IDS.

Vishwakarma and Kesswani (2022) proposed a Deep Neural Network-based IDS (DIDS) for the IoT environment. They used Neflow-based NIDS (NF-BoT-IoT, NF- ToN-IoT, NF-CSE CIC IDS2018, NF-UNSW NB15) datasets and the NF-UQ NIDSdataset, which is a combination of these four datasets. They compared the performance of the proposed DIDS model in multiclassification and binary classification. In binary classification, the highest accuracy was obtained with 99.21% in the NF-CSE CIC IDS2018 dataset, and in multiclassification with 97.48% in the NF-UNSW NB15 dataset.

Kim et al. (2029) the CNN and RNN DL models were utilized to detect DoS attacks. In their research, they utilized the CSE-CIC-IDS2018 and KDD CUP99 datasets to conduct binary and multiple classifications. At the conclusion of the study, the CNN model on the KDD CUP99 dataset obtained 99% accuracy in binary and multiclass classification, whereas the RNN model obtained 99% accuracy in binary classification and 93% accuracy in multiclassification. For the CSE-CIC-IDS 2018 dataset, the CNN model obtained an average accuracy of 91.5% while the RNN model achieved an average accuracy of 65%.

Sun et al. (2020) suggested hybrid research that utilized both CNN and LSTM. They developed a model that extracts the temporal and spatial properties of network traffic using the CICIDS 2017 dataset. This model achieved 98.67% performance in multiclassification.

Yang and Wang (2019) developed an enhanced CNN model for the NSL-KDD data set. After feature selection methods, they employed a total of 21 characteristics to detect four distinct types of attacks in their study. Accuracy, True Positive Rate (TPR), and False Positive Rate (FPR) were utilized to evaluate the suggested model. By the conclusion of the research, their accuracy was 95.36 percent.

Naveed et al. (2022) suggested a hybrid feature selection and Deep Neural Network (DNN) based classifier strategy for IDS. Methods of Principal Component Analysis, Chi-square and Analysis of Variance (ANOVA) were employed to produce a subset of characteristics that could be utilized for categorization The NSL-KDD dataset was utilized for the research. The suggested technique achieved 99.73% accuracy, 99.75% precision and 99.72% F1-score.

Table 1 provides a comparison of the studies that have been published.

Table 1 Comparative analysis of literature referenced studies

Reference	Dataset	Method	Attack Type	Results
Altunay and Albayrak (2021)	CSE-CIC-IDS2018	CNN+SMOTE	Botnet, SQ Injection, Brute Force, DoS	PRE: 99.5% REC: 98.1% ACC: 98.8%
Idrissi et al. (2021)	Bot-IoT	CNN, RNN, LSTM, GRU	Botnet	CNN: 99.94% RNN: 99.42% LSTM:99.74% GRU:99.43%
Vishwakarma and Kesswani (2022)		DIDS	Various types of IoT network attacks	**Binary Classification** **NF-BoT-IoT** ACC:99.08% PRE:99.03% REC:99.08% FSC: 99.02% **NF-ToN-IoT** ACC:99.48% PRE:99.48%

				REC:99.48% FSC: 99.48%
	NF-BoT-IoT, NF-ToN-IoT, NF-CSE CIC IDS2018, NF-UNSW NB15 NF-UQ NIDS			**NF-CSE CIC IDS2018** ACC:99.21% PRE:99.21% REC:99.21% FSC: 99.20% **NF-UNSW NB15** ACC:98.72% PRE:98.69% REC:98.72% FSC: 98.70% **NF-UQ NIDS** ACC:98.23% PRE:98.23% REC:98.23% FSC: 98.22%
Kim et al. (2020)	CSE-CIC-IDS2018, KDD CUP99	CNN, RNN	DoS, DDoS	**Binary Classification** **KDDCUP99/CNN** ACC:99% **KDD CUP99/RNN** ACC:99% **Multi Classification** **KDDCUP99/CNN** ACC:99% **KDD CUP99/RNN** ACC:93% **CSE-CIC-IDS 2018 CNN:** average 91,5% **CSE-CIC-IDS 2018RNN:** average 65%
Sun et al. (2020)	CICIDS2017	CNN+LSTM	FTP-Patator, SSH-Patator, DoS, Heartblee, Web Attack, Infiltration, Botnet, DDoS	ACC: 98.67% REC:97.21% PRE:0.47% FSC:93.32%
Yang and Wang (2019)	NSL-KDD CUP	Improved CNN	Probe,DoS R2L U2R	ACC:%95.36 PRE:%0.76 REC:%95.55
Naveed et al. (2020)	NSL-KDD	DNN	Normal Anomaly	ACC=%99,73 PRE=%99,75 REC=%99,73 FSC=%99,72

Note: ACC = Accuracy, PRE = Precision, REC = Recall, FSC = F1-Score

3. Material and Method

3.1 Dataset

This study utilized a NetFlow based structure of the BoT-IoT dataset to create the NF-BoT-IoT dataset (Sarhan et al., 2020). Cisco developed NetFlow in 1996, and it has been used to describe network flows. There are 600,100 samples in all. This includes 586,241 (97.69%) attack patterns and 13,859 (2.31%) normal flow patterns (benign). The dataset has twelve characteristics (Sarhan et al., 2020). Table 2 shows the characteristics of this data set.

Table 2 Characteristics of NF-BoT-IoT data set.

Characteristic	**Characteristic**
Description	**Description**
IPV4_SRC_ADDR	IPv4 source address
IPV4_DST_ADDR	IPv4 destination address
L4_SRC_PORT	IPv4 source port number
L4_DST_PORT	IPv4 destination port number
PROTOCOL	IP protocol identifier byte
TCP_FLAGS	Cumulative of all TCP flags
L7_PROTO	Layer 7 protocol
IN_BYTES	Incoming number of bytes
OUT_BYTES	Outgoing number of bytes
IN_PKTS	Incoming number of packets
OUT_PKTS	Outgoing number of packets
FLOW_DURATION_MILLISECONDS	Flow duration in milliseconds

3.2 Preprocessing

One of the most crucial factors determining how well DL models function is whether the data sets utilized are valuable and presented in a certain style (Tsimenidis et al., 2022). Data preprocessing or preparation is the transformation of data into a more usable format by employing techniques such as filling in missing values, finding, and cleaning outliers, eliminating duplicate data, combining data, and reducing the dimension data. This data's preprocessing in other words data preparation consisted of three phases. These phases are data cleaning, normalization and resampling.

Data cleaning is based on identifying missing, incorrect, or irrelevant parts of the data, and then replacing or deleting these parts (Chu et al., 2016). During the data cleaning phase of this research, four characteristics (Soderstrom, 2021; Wang et al., 2021) that were found ineffective for identifying network attacks were removed from all dataset entries, namely IPV4 SRC ADDR, L4 SRC PORT, IPV4 DST ADDR, and L4 DST PORT.

The second step of preprocessing involves data normalization, which is a technique used to standardize diverse data qualities (Aldallal, 2022). This involves rescaling the data within the range of 0 to 1. In this investigation, the normalization process was performed using the minimum-maximum formula described in Equation 1.

$$x_s = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (1)$$

In Equation 1, x_{max} and x_{min} represent the maximum and minimum values of the variable, respectively. When x equals the minimum value of the variable, the numerator becomes 0, so X_s equals 0. On the other hand, when x equals the maximum value of the variable, the numerator becomes equal to the denominator, resulting in X_s equaling 1. However, if x is between the minimum and maximum values, X_s takes a value between 0 and 1.

The third step of preprocessing is resampling. Class imbalance is a significant factor that affects the quality of classification performance, but it is often overlooked (Cengiz and Harman, 2022). Many justifications for using classification techniques only consider well balanced training sets, but this default balanced distribution may not exist in many datasets. One class may have very few instances, while the other may have a lot, leading to possible complications during the categorization phase. This can cause problems during the classification stage, as models may make erroneous predictions when presented with samples that have little label information due to inadequate training (Bedi et al., 2021). To mitigate the negative impact of training on imbalance datasets, three main methods are commonly used. These methods are oversampling, undersampling and synthetic data generation.

Oversampling involves duplicating samples from a randomly selected portion of the data belonging to the minority class until an even class distribution is obtained. However, this can increase the probability of overfitting because

it copies examples from the minority class. Undersampling, on the other hand, aims to rebalance the dataset by deleting a randomly selected portion of the samples from the majority class until the class distributions are equal. The SMOTE method is an oversampling process that enables the production of synthetic data by generating new minority class instances through certain operations between instances of the minority class. Synthetic samples are produced as follows:

1. A random sample is selected from the minority class and its k-nearest neighbor is found.
2. The difference value between the randomly selected sample and its k-nearest neighbor from the minority class is calculated.
3. The calculated difference value is multiplied by a random number (δ) between 0 and 1.
4. New samples are created according to Equation 2.

$$E_{new} = E_0 + (E_1 - E_0) \times \delta \qquad (2)$$

5. For each new data point, the first four steps are repeated.

Table 3 presents the distributions of attack and benign flow classes in the dataset before and after resampling.

Table 3 Distributions of attack and benign flow classes in the dataset

	Total number of data flows	**Number of attack flows**	**Number of normal flows**
Data distribution of the dataset	600.100	586.241	13.859
Data distribution after SMOTE	1.172.482	586,241	586,241
Data distribution after random over sampling	1.172.482	586,241	586,241
Data distribution after random undersampling	27.718	13.859	13.859

3.3 Convolutional Neural Network

Convolutional Neural Networks (CNN) are a DL method developed by LeCun in the LeNet architecture in 1998 (LeCun et al., 1998). CNNs are commonly used for analyzing visual information, such as image recognition and classification, natural language processing, medical image analysis, and speech analysis. The CNN model consists of three main layers: Convolutional layer, pooling layer and fully connected layer. The convolutional layer is the first layer in CNN algorithms. In this layer, the input data is passed through a filter to create a feature map. The second layer after the convolutional layer is the pooling layer, which is typically applied to the feature matrices created by the convolutional layer. Like the convolutional layer, the pooling layer aims to reduce dimensionality (Çetiner, 2021). This reduces the required computational power and focuses on more important features by ignoring unnecessary ones. The fully connected layer works on an input where each entry is connected to all neurons. This layer is typically found towards the end of the CNN architecture and is used to optimize goals such as class scores. The CNN structure used in this study is shown in Figure 1.

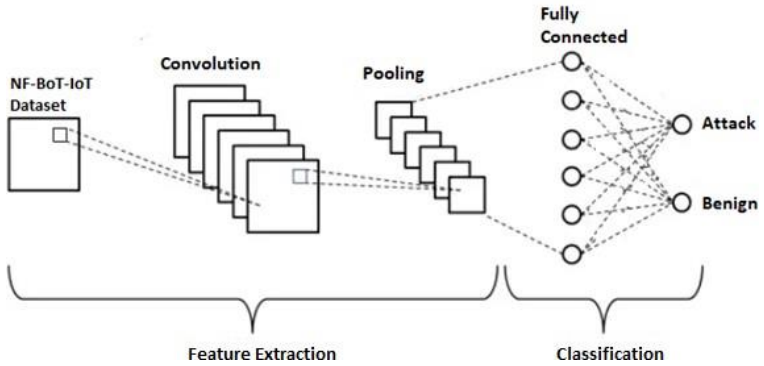

Figure 1 CNN structure

The variables whose values vary depending on the nature of the problem and the dataset in question are referred to as hyperparameters. How to construct a multi interactive Artificial Neural Network (ANN), as

well as the hyperparameter valuesutilized in the developed or created model, is a crucial aspect to consider when trying to find a solution to a problem using DL methods. The learning rate, number of neurons, epoch number, batch size, dropout, activation and optimization function usage are the hyper parameters that are often included in CNN constructions.
The hyperparameter values selected for training are given in Table 4.

Table 4 Hyper parameter values

Hyperparameter	Values
Learning rate	0.001
Activation Function	Relu
Optimization Method	Adam
Epoch number	10
Dropout	0.5
Number of neurons	32

The recommended flowchart for this model is given in Figure 2.

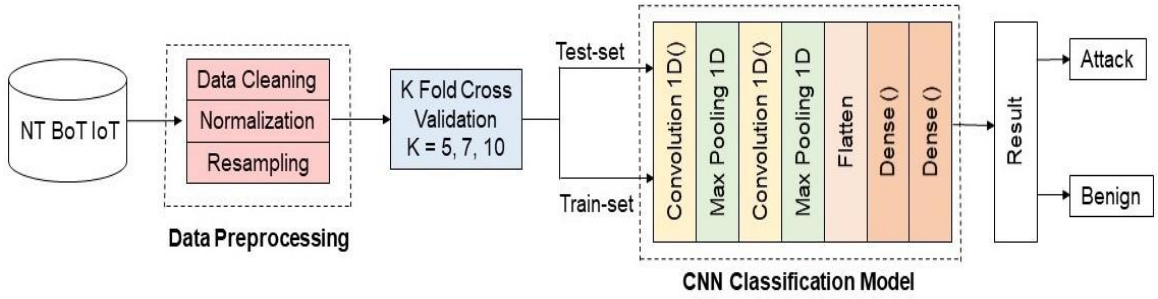

Figure 2 Flow chart of the proposed model

In this flow diagram, we demonstrate the data preprocessing and classification process using a CNN model. Initially, data were obtained from the NF BoT IoT dataset, which contains information related to IoT network traffic. The preprocessing stage consisted of three steps: data cleaning, normalization, and resampling. Data cleaning ensured that the dataset was free from inconsistencies or irrelevant information. With normalization, we scaled the data to a standard range to enhance the performance and convergence of the model. Resampling techniques were applied to address class imbalances in the dataset. After preprocessing, the dataset underwent K-fold cross validation with K values of 5, 7, and 10. This method divides the data into K subsets, where, in each iteration, one subset is used as the test set, and the remaining subsets are used as the training set. This allowed for a comprehensive evaluation of the generalization ability of the model. The pre processed and validated data were then fed into the CNN classification model. This model consists of two one dimensional convolution layers followed by max pooling layers. These layers extract fundamental features from the data and perform downsampling. The resulting feature maps were flattened and passed through two dense (fully connected) layers that facilitated the learning of complex patterns and decision boundaries. The output from the dense layers classifies the network traffic into two categories: 'Attack' and 'Benign.' Our study provides a classification that distinguishes between malicious and normal network activities, aiding the identification and mitigation of potential IoT security threats.

4. Experiment

4. 1 Evaluation Metrics

The study investigates the implications of the algorithm's performance by selecting the K value to be either 5, 7, or 10. To prevent an imbalanced distribution ofclasses within the dataset, the SMOTE, Random Under Sampling, and Random Over Sampling approaches were utilized. The performance results were analyzed by applying various K values to each of these approaches and comparing the outcomes. To assess the efficacy of the classification models, a confusion matrix was used as a comparison tool between the

estimated values of the target characteristic and the actual values. The confusion matrix is shown in Table 5.

Table 5 Confusion Matrix

		Predicted	
		Negative	Positive
Actual	Negative	**True Negative (TN)**	**False Positive (FP)**
	Positive	**False Negative (FN)**	**True Positive (TP)**

True Positive (TP): The situation where an attack instance is correctly classified as an attack.
False Positive (FP): The situation where a benign is incorrectly classified as an attack instance.
False Negative (FN): False Negative (FN): The situation where an attack instance is incorrectly classified as benign.
True Negative (TN): The situation where a benign is correctly classified as a benign.
To evaluate the performance of the models, accuracy, sensitivity, precision, and F1-score metrics were used. These values are calculated according to Equations [3-6].
Accuracy: The percentage of positive classified examples.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \qquad (3)$$

Precision: The proportion of true positives among the samples predicted as positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad (4)$$

Recall (sensitivity): A metric that shows how many of the positive values that we should have predicted as positive have actually been predicted as positive.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (5)$$

F1 score: The harmonic mean of precision and recall values.

$$\text{F}_1 = 2 * \frac{\text{precision}*\text{recall}}{\text{precision+recall}} \qquad (6)$$

4.2 Experimental Results

In this section, the confusion matrix and performance results are provided and discussed. Table 6 presents the confusion matrix obtained according to the resampling methods and K values.

Table 6 Confusion matrix obtained according to the resampling methods and K values.

Confusion Matrix for SMOTE

K=5		Prediction Value	
		negative	positive
Actual Value	negative	69068	48180
	positive	5014	112234

K=7		Prediction Value	
		negative	positive
Actual Value	negative	49262	34486
	positive	3534	80215

K=10		Prediction Value	
		negative	positive
Actual Value	negative	34087	24537
	positive	2501	56123

Confusion Matrix for Random Over Sampling

K=5		Prediction Value	
		negative	positive
Actual Value	negative	68983	48265
	positive	5014	112234

K=7		Prediction Value	
		negative	positive
Actual Value	negative	49274	34474
	positive	3534	80215

K=10		Prediction Value	
		negative	positive
Actual Value	negative	34438	24186
	positive	2501	56123

Confusion Matrix for Random Under Sampling

K=5		Prediction Value	
		negative	positive
Actual Value	negative	1625	1146
	positive	127	2645

K=7		Prediction Value	
		negative	positive
Actual Value	negative	1166	813
	positive	84	1896

K=10		Prediction Value	
		negative	positive
Actual Value	negative	799	586
	positive	71	1315

As shown in Table 6, when the SMOTE technique was applied, the TP and TN were high for K=5, K=7, and K=10. The highest TP and TN values were obtained for K=5, indicating that the model generally performed better. For K=7 and K=10, we observed an increase in the FP and FN values, indicating that the performance of the model decreased slightly as the K value increased. For the random oversampling technique, the TP and TN values for K=5, K=7, and K=10 were quite similar to those obtained with SMOTE. The highest TP and TN values were observed at K=5, indicating that this method performed the best at K=5. We observed an increase in the FP and FN values for K=7 and K=10, indicating that the performance of the model decreased as the K value increased. Random undersampling reduces this imbalance by decreasing the number of examples in the majority class. The results showed that the TP and TN values for K=5, K=7, and K=10 were quite low, indicating that the dataset was significantly reduced and information loss occurred. Although the TP and TN values were obtained with K=5, these values were quite low compared those with of the other methods. For K=7 and K=10, we observed a slight increase in the FP and FN values, indicating that the performance of the model decreased further as the K value increased. Table 7 displays the accuracy and performance metrics of the CNN method applied to the dataset.

Table 7 Performance metrics of the CNN method

SMOTE

K Fold Cross Validation	Precision	Recall	F1-Score	Accuracy
K=5	0.70	0.96	0.81	0.77
K=7	0.70	0.96	0.81	0.77
K=10	0.70	0.96	0.81	0.77

Random Over Sampling

K=5	0.70	0.96	0.81	0.77
K=7	0.70	0.96	0.81	0.77
K=10	0.70	0.96	0.81	0.77

Random Under Sampling

K=5	0.70	0.95	0.81	0.77
K=7	0.70	0.96	0.81	0.77
K=10	0.69	0.95	0.80	0.76

Table 7 presents the performance results obtained using K-Fold Cross-validation for SMOTE, random oversampling, and random undersampling methods.We analyzed the precision, recall, F1-score, and accuracy for each method with K values of 5, 7, and 10. SMOTE and random oversampling methods yielded the same results for all three K values: a precision value of 0.70, a recall value of 0.96, an F1-score of 0.81, and an accuracy of 0.77. This indicates that both methods work similarly and provide a balanced performance. For the random under-sampling method, the precision value was 0.70, the recall value was 0.95 or 0.96, the F1-score was 0.81, and the accuracy was 0.77 for K=5 and K=7. However, for K=10, the precision value dropped to 0.69, recall value was 0.95, F1-score was 0.80, and accuracy was 0.76. These results suggest that the performance of the random undersampling method slightly decreases for K=10. Overall, it was observed that SMOTE and Random Over Sampling methods provide more consistent and reliable results, while the performance of the Random Under Sampling method varies depending on the K value.

Accuracy, precision, recall, and F1-score were some of the metrics used to assess the quality of the developed models. It was observed that these parameters produced identical outcomes. In our prior research on the NF-BoT-IoT dataset (Cengiz and Harman,2022), using Random Forest (RF), K-Nearest Neighbor (KNN), Support Vector Machines (SVM), and ANN binary classification was performed. In this investigation, an accuracy of 99.4 percent was achieved with RF, 82.7 percent with KNN, 96.7 percent with SVM, and 60.7 percent with ANN. When comparing the two experiments, it was shown that ANN had a higher accuracy. Figure 3, Figure 4 and Figure 5 shows the performance metrics of the CNN methods.

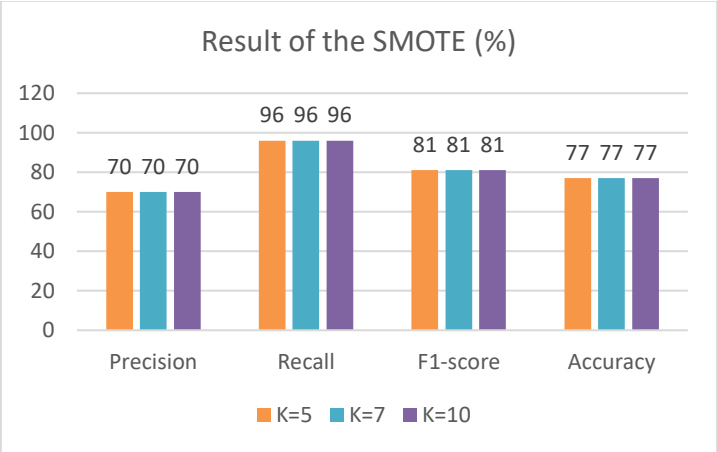

Figure 3. Performance metrics of the SMOTE.

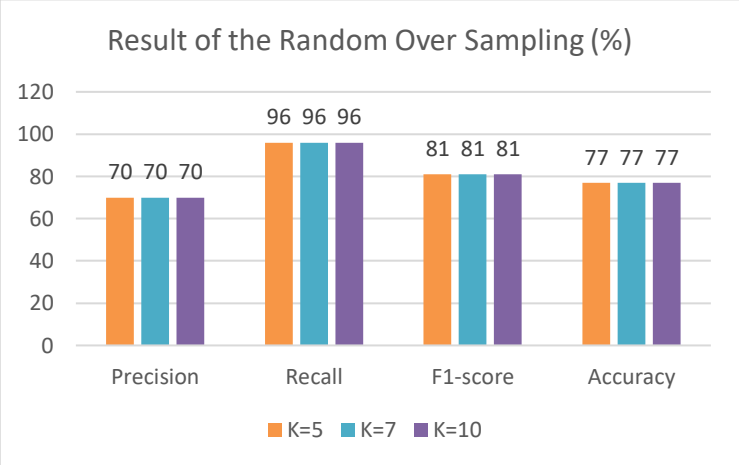

Figure 4. Performance metrics of the Random Over Sampling.

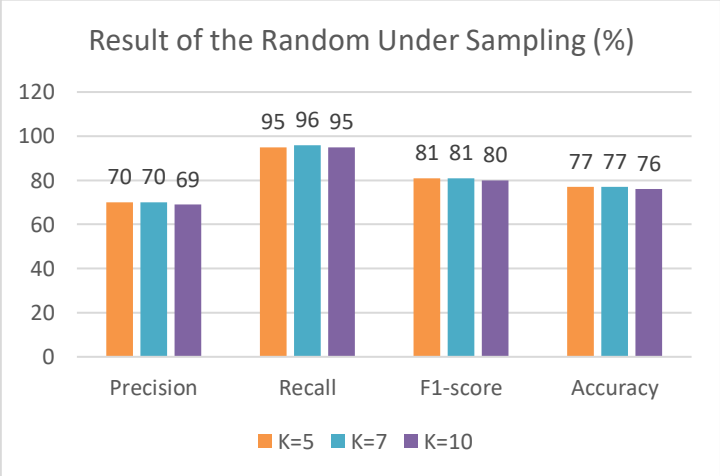

Figure 5. Performance metrics of the Random Under Sampling.

5. Conclusion

TCP/IP sends and receives data over the Internet for computers and other smart devices. Social media, online meetings, and applications are common. As financial difficulties like money transfers and electronic commerce are included all networks that receive, give, or engage the internet network must contain security. Internet technology simplifies many parts of life. Nonetheless, this situation has created huge risks. As security risks grow, so do security technology. Authentication and access control software stop unauthorized users from accessing or changing their access information by restricting access or prohibiting access entirely. Firewalls and anti-virus

software greatly reduce material and moral damages. Threats break this software. Real-time IDS protects device connectivity and detects network attacks. Network traffic and activity alert IDS to threats. Signature and anomaly-based IDSs exist. Anomaly based IDS informs network management. Signature-based IDS detects attacks by compar- ing connection data to the signature database. In this particular study, the objective is to identify network threats using the DL technique. The NF-BoT-IoT dataset has a skewed distribution of classes. To improve the model's performance, the imbalance ratio was decreased using SMOTE, Random Over Sampling, and Random Under Sam- pling techniques. The K Fold Cross Validation method was used to determine the success of the implemented procedures, with K values of 5, 7, and 10. The purpose of this research is to analyze how resampling and K-value affects output performance. The research found that both the SMOTE and Random Over Sampling techniques achieved accuracy within 77% across a range of K values. For K=5, 7, and 10, the Random Under Sampling technique achieved an accuracy of 77% and 76%. The outcomes were consistent across resampling strategies and K values. This means that as the K value for the NF-BoT-IoT dataset increases, so do the associated costs and time requirements.

Declarations

Ethical Approval The work follows appropriate ethical standards in conducting research and writing the manuscript. This work presents computational models trained with publicly available data, for which no ethical approval was required.

Competing interests The authors declare that they have no competing interests.

Authors contribution statement: The authors confirm their contribution to the paper as follows: study conception and design: Harman and Cengiz; data collection: Cengiz; analysis and interpretation of results: Harman and Cengiz; draft manuscript preparation: Harman and Cengiz. All authors reviewed the results and approved the final version of the manuscript.

Data availability and access The public dataset https://staff.itee.uq.edu.au/ marius/NIDS datasets/

References

Aldallal, A., 2022 Toward efficient intrusion detection system using hybrid deep learning approach. Symmetry, 14(9), 1916. https://doi.org/10.3390/ sym14091916

Altunay, H.C., Albayrak, Z., 2021. Network intrusion detection approach based on con- volutional neural network. Avrupa Bilim ve Teknoloji Dergisi, (26), 22–29. https://doi.org/10.31590/ejosat.954966

Baykan, N.A., Khorram, T., 2021. Network intrusion detection using optimized machine learning algorithms. Avrupa Bilim ve Teknoloji Dergisi, (25), 463–474. https://doi.org/10.31590/ejosat.849723

Baykara, M., Resul, D., 2019. Saldırı tespit ve engelleme araçlarının incelenmesi. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(1), 57–75. https://doi.org/10.24012/dumf.449059

Bedi, P., Gupta, N., Jindal, V., 2021. I-siamids: an improved siam-ids for handling class imbalance in network-based intrusion detection systems. Applied Intelligence, 51, 1133–1151. https://doi.org/10.1007/s10489-020-01886-y

Behera, S., Pradhan, A., Dash, R., 2018. Deep neural network architecture for anomaly based intrusion detection system. In: 2018 5th International Conference on SignalProcessing and Integrated Networks (SPIN), pp. 270–274. https://doi.org/ 10.1109/SPIN.2018.8474162

Butun, I., Morgera, S.D., Sankar, R., 2013. A survey of intrusion detection systemsin wireless sensor networks. IEEE communications surveys & tutorials, 16(1), 266–282. https://doi.org/10.1109/SURV.2013.050113.00191

Cengiz, E., Harman, G., 2022 Dengesiz ml-tabanlı nıds veri setlerinin sınıflandırma performanslarının karşılaştırılması. Avrupa Bilim ve Teknoloji Dergisi, (41), 349–356. https://doi.org/10.31590/ejosat.1157441

Çetiner, H., 2021. Classification of apple leaf diseases using the proposed convolution neural network approach. Mühendislik Bilimleri Ve Tasarım Dergisi, 9(4), 1130-1140. https://doi.org/10.21923/jesd.980629

Gülcü, A., Kuş, Z.,2019. A survey of hyper-parameter optimization methods in con- volutional neural networks. Gazi Üniversitesi Fen Bilimleri Dergisi, 7(2), 503–522. https://doi.org/10.29109/gujsc.514483

Idrissi, I., Boukabous, M., Azizi, M., Moussaoui, O., El Fadili, H., 2021. Toward a deep learning-based intrusion detection system for iot against botnet attacks. IAES International Journal of Artificial Intelligence, 10(1), 110. https://doi.org/10.11591/ijai.v10.i1.pp110-120

Ilyas, I.F., Chu, X., 2019. Data Cleaning. Morgan Claypool.

Kim, J., Kim, J., Kim, H., Shim, M., Choi, E., 2020. Cnn-based network intrusiondetection against denial-of-service attacks. Electronics, 9(6), 916. https://doi.org/10.3390/electronics9060916

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 https://doi.org/10.48550/arXiv.1412.6980

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.https://doi.org/10.1109/5.726791

Mandal, D., Kösesoy, İ., 2023. Prediction of Software Security Vulnerabilities from Source Code Using Machine Learning Methods. In 2023 Innovations in Intelligent Systems and Applications Conference (ASYU), pp. 1-6, IEEE. https://doi.org/10.1109/ASYU58738.2023.10296747

Mijalkovic, J., Spognardi, A., 2022. Reducing the false negative rate in deep learningbased network intrusion detection systems. Algorithms, 15(8), 258. https: //doi.org/10.3390/a15080258

Naveed, M., Arif, F., Usman, S.M., Anwar, A., Hadjouni, M., Elmannai, H., Hussain, S., Ullah, S.S., Umar, F., 2022. A deep learning-based framework for feature extraction and classification of intrusion detection in networks. Wireless Communications and Mobile Computing, 2022. https://doi.org/10.1155/2022/ 2215852

Otoum, Y., Nayak, A.,2021. As-ids: Anomaly and signature based ids for the internetof things. Journal of Network and Systems Management, 29, 1–26.

Priyadarshini, R., Barik, R.K., 2022. A deep learning based intelligent frameworkto mitigate ddos attack in fog environment. Journal of King Saud University- Computer and Information Sciences, 34(3), 825–831 (2022). https://doi.org/10.1016/j.jksuci.2019.04.010

Söderström, A., 2021. Anomaly-based Intrusion Detection Using Convolutional Neural Networks for IoT Devices.

Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M., 2020. Netflow datasets for machine learning-based network intrusion detection systems. In: Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and13th EAI International Conference on Wireless Internet, WiCON 2020, Vir-tual Event, December 11, 2020, Proceedings 10, pp. 117–135. https://doi.org/10.1007/978-3-030-72802-1 9

Sarkar, S., Chatterjee, S., Misra, S., 2015. Assessment of the suitability of fog computing in the context of internet of things. IEEE Transactions on Cloud Computing, 6(1),46–5. https://doi.org/10.1109/TCC.2015.2485206

Shone, N., Ngoc, T.N., Phai, V.D., Shi, Q., 2018. A deep learning approach to net- work intrusion detection. IEEE transactions on emerging topics in computational intelligence, 2(1), 41–50. https://doi.org/10.1109/TETCI.2017.2772792

Sun, P., Liu, P., Li, Q., Liu, C., Lu, X., Hao, R., Chen, J., 2020. Dl-ids: Extracting features using cnn-lstm hybrid network for intrusion detection system. Security and communication networks, 2020, 1–11. https://doi.org/10.1155/2020/ 8890306

Tsimenidis, S., Lagkas, T., Rantos, K., 2022. Deep learning in iot intrusion detection. Journal of network and systems management, 30, 1–40.

Vishwakarma, M., Kesswani, N., 2022. Dids: A deep neural network based real-time intrusion detection system for iot. Decision Analytics Journal, 5, 100142. https://doi.org/10.1016/j.dajour.2022.100142

Vishwakarma, M., Kesswani, N., 2022. Dids: A deep neural network based real-time intrusion detection system for iot. Decision Analytics Journal, 5, 100142. https://doi.org/10.1016/j.dajour.2022.100142

Wang, C., Wang, B., Sun, Y., Wei, Y., Wang, K., Zhang, H., Liu, H., 2021. Intrusion detection for industrial control systems based on open set artificial neural network. Security and Communication Networks, 2021, 1–14. https://doi.org/10. 1155/2021/4027900

Yang, H., Wang, F., 2019. Wireless network intrusion detection based on improvedconvolutional neural network. Ieee Access, 7, 64366–64374. https://doi.org/10.1109/ACCESS.2019.2917299