Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.050586

ARTICLE

CNN Channel Attention Intrusion Detection System Using NSL-KDD Dataset

Fatma S. Alrayes[1], Mohammed Zakariah[2], Syed Umar Amin[3,*], Zafar Iqbal Khan[3] and Jehad Saad Alqurni[4]

[1]Information Systems Department, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, 11671, Saudi Arabia

[2]College of Computer and Information Sciences, King Saud University, Riyadh, 11362, Saudi Arabia

[3]College of Computer and Information Sciences, Prince Sultan University, Riyadh, 11586, Saudi Arabia

[4]Department of Educational Technologies, College of Education, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam, 31441, Saudi Arabia

*Corresponding Author: Syed Umar Amin. Email: samin@psu.edu.sa

Received: 11 February 2024 Accepted: 15 April 2024 Published: 20 June 2024

ABSTRACT

Intrusion detection systems (IDS) are essential in the field of cybersecurity because they protect networks from a wide range of online threats. The goal of this research is to meet the urgent need for small-footprint, highly-adaptable Network Intrusion Detection Systems (NIDS) that can identify anomalies. The NSL-KDD dataset is used in the study; it is a sizable collection comprising 43 variables with the label's "attack" and "level." It proposes a novel approach to intrusion detection based on the combination of channel attention and convolutional neural networks (CNN). Furthermore, this dataset makes it easier to conduct a thorough assessment of the suggested intrusion detection strategy. Furthermore, maintaining operating efficiency while improving detection accuracy is the primary goal of this work. Moreover, typical NIDS examines both risky and typical behavior using a variety of techniques. On the NSL-KDD dataset, our CNN-based approach achieves an astounding 99.728% accuracy rate when paired with channel attention. Compared to previous approaches such as ensemble learning, CNN, RBM (Boltzmann machine), ANN, hybrid auto-encoders with CNN, MCNN, and ANN, and adaptive algorithms, our solution significantly improves intrusion detection performance. Moreover, the results highlight the effectiveness of our suggested method in improving intrusion detection precision, signifying a noteworthy advancement in this field. Subsequent efforts will focus on strengthening and expanding our approach in order to counteract growing cyberthreats and adjust to changing network circumstances.

KEYWORDS

Intrusion detection system (IDS); NSL-KDD dataset; deep-learning; machine-learning; CNN channel Attention; network security

Abbreviations

IDS	Intrusion detection system
CNN	Convolutional neural network

RBM	Restricted boltzmann machine
IoT	Internet of things
SIEM	Security information and event management
ML	Machine learning
DNN	Deep neural network
RF	Random forest
KNN	K-nearest neighbor
EBT	Ensemble boosted learning
TCP	Transmission control protocol
ICMP	Internet control message protocol
R2L	Remote to local
ECA	Efficient channel attention
CBAM	Convolutional block attention module
ROC	Receiver operating characteristics
TN/TP	True negative/true positive
AUC	Area under the curve
FN/FP	False negative/false positive
NIDS	Network intrusion detection system
ANN	Artificial neural network
MCNN	Multiclass CNN
HIDS	Host-based IDS
SIDS	Signature-based IDS
DL	Deep learning
SVM	Support vector machine
DT	Decision tree
PCA	Principle component analysis
IDN	Intrusion detection network
UDP	User datagram protocol
U2R	User to root
DoS	Denial of services
GAP	Global average pooling
ROC	Receiver operating characteristics

1 Introduction

Amid the rapidly evolving landscape of cyber threats, network intrusion detection plays a vital role in protecting against advanced attacks [1,2]. The traditional reliance on feature detection in intrusion detection systems (IDS) has shown effectiveness but is encountering increasing challenges [3,4]. The efficacy of IDS that rely on existing attack patterns to identify novel forms of attacks is constrained by the finite capacity to update databases with such patterns. Researchers have recognized this approach's constraints and shifted their focus towards using sophisticated methods [5]. Machine learning has become a practical approach for enhancing the capabilities of intrusion detection systems. Moreover, the growing usage of computer networks and the integration of the Internet of Things (IoT) in industrial environments, combined with a diverse set of applications running on these networks, have heightened network security vulnerabilities [6]. The escalation of hazards has garnered the attention of researchers and data scientists, spurring them to seek innovative approaches to enhance security

systems [7]. Meanwhile, the KDD99 dataset and its subsequent refinement, the NSL-KDD dataset, are crucial benchmarks for developing and evaluating network-based IDSs.

Further, an IDS is essential for safeguarding the integrity of a network. The system consistently monitors network data transmission, identifying and alerting administrators of any abnormal behavior that may indicate a security breach [7,8]. The IDS operate as a software application that conducts scans across the network or system to detect malevolent activity or violations of regulations [9,10]. Any identified malicious behavior is promptly reported to the system administrator or consolidated centrally using a Security Information and Event Management (SIEM) system. The SIEM system employs several data sources and incorporates alert filtering methods to distinguish genuine security threats from false positives.

Furthermore, within the realm of IDS, there exist two primary categorizations: Host-based IDS (HIDS) and network-based IDS (NIDS) [11–13]. The HIDS is strategically deployed on specific endpoints to provide comprehensive security measures against potential hazards from internal and external sources. This specific IDS can observe the movement of network data to and from the host machine, meticulously inspect the running processes, and scrutinize the system logs [14,15].

In contrast, NIDS is specifically designed to oversee and manage an entire network, offering extensive understanding of all the data traversing the network [16,17]. The decision-making process in NIDS is based on both packet information and contents, which allows for a more extensive perspective that facilitates the detection of widespread threats [18,19]. However, NIDS cannot monitor the internal processes of the individual endpoints it protects.

In addition, signature-based IDS employ pre-defined patterns, called signatures, to detect and identify malicious attacks [20,21]. These patterns can be comprised of specific byte sequences or the presence of identified malicious instructions in network traffic [22,23]. Signature-based intrusion detection systems (IDS) identify attacks with documented signatures [24–26]. Nevertheless, they have challenges discerning unfamiliar malware attacks that have not been previously encountered, mostly due to the absence of a pre-existing template for comparison. Conversely, anomaly-based IDS bypasses this limitation by employing machine learning to construct a reliable model of network behavior [27–29]. The incoming data is subsequently compared to this model, and any departures from the expected behavior are detected as suspicious. The anomaly-based IDS that utilizes machine learning is characterized by its superior flexibility compared to signature-based systems [30]. This is due to the ability of the models to be trained to operate well over a broader spectrum of applications and hardware configurations.

In the realm of IDS, traditional CNNs have long been effective. However, as cyber threats evolve in sophistication, the demand for more nuanced detection mechanisms arises. While CNNs excel at capturing spatial correlations within data, they often overlook crucial inter-channel relationships, limiting their ability to detect subtle harmful patterns. This shortfall underscores the necessity for enhanced models that leverage geographical information and integrate methods to identify and prioritize salient elements across channels. Dynamic feature map adjustments through channel attention techniques offer a promising solution to this challenge. By incorporating such methods into CNN designs, discriminative capacity is heightened, enabling a focus on pertinent information while minimizing noise, consequently enhancing IDS detection accuracy. Our research delves into the fusion of CNN with channel attention for intrusion detection, aiming to bolster IDS capabilities in combatting contemporary cyber threats. Moreover, in the landscape of IDS and machine learning research, the KDD99 dataset has maintained its significance despite its age of over 15 years. Nonetheless, the emergence of the NSL-KDD dataset as a refined alternative has garnered attention. Our study

undertakes a comprehensive analysis of a NSL-KDD datasets. While KDD99 remains prevalent, NSL-KDD exhibits advancements in IDS and ML functionality.

Above, Fig. 1 shows the CNN Channel Attention Intrusion Detection System (IDS) procedure using the NSL-KDD dataset. After data preparation, feature extraction extracts useful information from network traffic for categorization and detection. As described in the methodology and model evaluation sections, the trained model is tested using distinct data. During detection, the system generates IDS alarms for unusual network traffic patterns, alerting potential intrusions. These sirens notify security staff to investigate and neutralize hazards. Accuracy, precision, F1-score, recall, and ROC curves are used to evaluate parameter testing findings, which optimize system performance. These measurements show how well the system distinguishes between regular and malicious network activity, determining intrusion detection success. This systematic approach helps the IDS protect network infrastructure from security breaches, improving cybersecurity and threat mitigation.

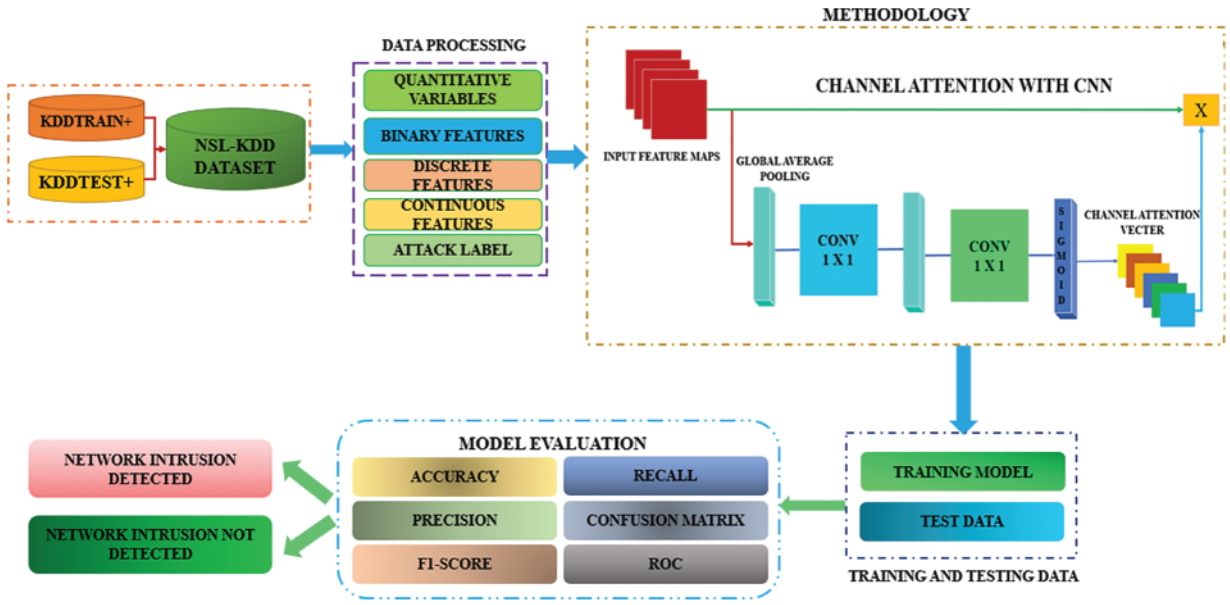

Figure 1: CNN channel attention intrusion detection system framework implemented with the NSL-KDD dataset

The contributions of this research paper are as follows:

i) Enhanced NSL-KDD dataset validation showcases its relevance, aiding in evaluating intrusion detection systems under various attack scenarios.
ii) Integration of CNN with channel attention techniques boosts detection accuracy, meeting the evolving demands of cybersecurity.
iii) Adaptable solution addresses dynamic cybersecurity threats, aligning with the evolving nature of intrusion detection.
iv) Contributes innovative intrusion detection system, improving feature selection and response to cyber threats.

In addition, Section 2 examines pertinent literature, investigating prior research in the topic. Subsequently, Section 3 elucidates the data collecting procedure, whereas Section 4 delineates the

selected methodology. The collected results are presented in Section 5, and a full explanation is offered in Section 6. The review is concluded in Section 7, where major findings and ideas are summarized.

2 Literature Review

The domain of intrusion detection systems (IDS) has witnessed a noteworthy surge in the creation of innovative methodologies to augment the precision of threat identification [1–3]. Using the NSL-KDD dataset as an example, this article examines the applications of convolutional neural networks (CNN) and channel attention mechanisms in IDS [4,5]. Using the NSL-KDD dataset, eight eminent researchers have thoroughly examined and improved intrusion detection methods, leading to notable advancements in the area [6]. This work aims to improve threat detection capabilities by introducing a CNN Channel Attention Intrusion Detection System. This literature review lays the basis for this study.

Convolutional neural network-based intrusion detection systems (IDSs) were studied by Mohammadpour et al. [7]. Emerging issues in network security are discussed in this article. Understanding CNN's many applications to identify network breaches and attacks is emphasized in the introduction. Effectiveness is a prerequisite for intrusion detection systems (IDSs) to defend against evolving cyber threats. The KDDCup99, NSL-KDD, UNSW-NB15, and CIC-IDS2017 databases are used for this. There are 22,544 testing samples and 125,973 training samples in NSL-KDD. CNN-based IDS techniques are categorized using a brand-new system. The study found that the hybrid auto-encoder with CNN achieved 84.39% accuracy on the NSL-KDD dataset [8,9].

Furthermore, Al-Turaiki et al.'s research [10] presents a novel anomaly-based network intrusion detection method. For deep feature synthesis and dimensionality reduction, they employ a CNN. The investigation exposes a significant vacuum in the body of knowledge regarding the model's capacity to expand and be stable under various network conditions. It is challenging to extrapolate the results to datasets other than benchmark examples. There are 125,973 and 22,544 records in the NSL-KDD and UNSW-NB15 datasets, respectively. The KDDTest+ and KDDTrain+ subsets are assessed in the study. The technique produces deep features through feature engineering and dimensionality reduction [11,12].

To assess IDSs, the KDDCup99 and NSL-KDD datasets are frequently utilized. Sapre et al. [13] stress the importance of examining these datasets in-depth. More classification measures and several machine learning (ML) classifiers are used in this study than in previous evaluations. The NSL-KDD dataset performs better, with an average accuracy advantage of 20.18% over KDDCup99. The 67,343 normal sample NSL-KDD dataset is assessed using ANN and further machine learning techniques. The ANN outperforms the average with an accuracy of 78.51% for the NSL-KDD dataset.

Similarly, investigating sophisticated methods for identifying unauthorized access is crucial to cybersecurity in the digital age. The NSL-KDD dataset is used in this study by Gao et al. [17] to address intrusion detection technology difficulties. Accurate detection by adaptive ensemble learning is the main focus of the work. Multiple decision trees and modified training data proportions are employed in the MultiTree approach. The work employs ensemble adaptive voting with decision trees, random forests, kNN, and DNN to increase detection accuracy. Positive outcomes from validating the NSL-KDD Test+ dataset demonstrate that the MultiTree algorithm attains 84.2% accuracy and 85.2% adaptive voting. Compared to earlier studies, the ensemble model increases detection accuracy [18,19]. This illustrates how detection results are impacted by data quality.

Furthermore, Magdy et al. [20] used the NSL-KDD dataset to test intrusion detection systems (IDS). Decision Tree, CNN, Random Forest, SVM, KNN, and ANN intrusion detection techniques are all compared in the study. The topic is how to effectively use IDS to improve network security. The researchers used NSL-KDD, including 22,544 test records and 125,973 training records, and UNSW-NB15. The work employs the NSL-KDD dataset; however, it extensively uses machine learning techniques. Applying the results to other network configurations can be more difficult. Take note that the CNN model's accuracy on NSL-KDD was 79.48%. It is demonstrated how successful several algorithms are in this intrusion detection scenario.

Moreover, Imamverdiyev et al. [23] improve cybersecurity by researching the use of deep learning—specifically, the Restricted Boltzmann Machine (RBM)—to identify denial-of-service (DoS) threats. The NSL-KDD dataset is used in the study's evaluation of the suggested methodology. This research aims to increase detection accuracy to counteract denial-of-service (DoS) assaults, a problem in cyber security. The study also examines RBM, deep learning, and SVM [24–26]. The study's limited application to other cyber-threat scenarios may stem from its sole focus on improving the precision of DoS attacks. The RBM model's accuracy on the NSL-KDD dataset is 73.23%, demonstrating how successfully the deep learning method tackles cybersecurity issues.

In addition, Zakariah et al. [27] added cybersecurity expertise by examining an intrusion detection system (IDS) using tailored machine learning techniques on the NSL-KDD dataset. Based on the dataset, this study employs 22,544 testing samples and 125,973 training samples (KDDTrain+). This research focuses on using intrusion detection to improve network security. ANN and PCA are used in the research to mimic intricate network architectures. Despite the ANN model's 97.5% accuracy, the study admits its reliance on the NSL-KDD dataset is a constraint. Due to this limitation, the proposed IDS might be challenging to modify for various network conditions and datasets [28–30].

To strengthen IoT cybersecurity, Abu Al-Haija et al. [31] also assessed machine-learning-based network intrusion detection systems (NIDSs). Ensemble-boosted trees (EBT) are used in the assessment. The paper tackles the pressing problem of protecting IoT networks from malevolent attackers. Because the study only examined the NSL-KDD and distilled Kitsune-2018 datasets, its findings might not generalize to other IoT networks. A large NSL-KDD dataset including 67,313 training and 9,711 testing samples is used to train the method. Testing the suggested strategy in real-world settings is essential. Ensemble Boosted Trees achieve 99.1% accuracy on the NSL-KDD dataset, promising results for IoT network traffic security [32–36].

Table 1 shows the list of past references including datasets, methodology, limitations and results.

Table 1: List of past references including datasets, methodology, limitations and results

Ref.	Dataset	Methodology	Limitations	Results
[7]	NSL-KDD	Deep-learning, LSTM, RNN, hybrid autoencoder with CNN	Absence of real-world validation, potential CNN bias, reliance on existing datasets.	84.39% accuracy with hybrid autoencoder and CNN.

(Continued)

Table 1 (continued)

Ref.	Dataset	Methodology	Limitations	Results
[10]	NSL-KDD, UNSW-NB15	Intrusion detection, CNN, BCNN, multi-class CNN	Lacks scalability exploration, limited to benchmark datasets.	MCNN achieves 99.5% accuracy on NSL-KDD.
[13]	KDDCup99, NSL-KDD	ML classifiers, ANN	Focus on accuracy metrics, overlooking other aspects, dataset generalization.	ANN achieves 78.51% accuracy for NSL-KDD.
[17]	NSL-KDD, UNSW-NB15	Adaptive ensemble model, multi-tree algorithm, DNN	Limited insight on scalability, real-world implementation challenges.	Adaptive voting: 85.2%, multi-tree: 84.2% accuracy.
[20]	NSL-KDD, UNSW-NB15	Machine learning, decision tree, CNN, Random Forest, SVM, KNN, ANN	Focus on NSL-KDD restricts applicability, neglects other datasets.	CNN model attains 79.48% accuracy on NSL-KDD.
[23]	NSL-KDD	Deep-learning method, RBM, SVM	Concentration on DoS attack detection, limited applicability.	RBM achieves 73.23% accuracy with NSL-KDD.
[27]	NSL-KDD	PCA, ANN	Sole focus on NSL-KDD limits IDS applicability.	ANN achieves 97.5% accuracy on NSL-KDD.
[31]	NSL-KDD	Ensemble-learning, EBT	Limited to specific datasets, restricts conclusions' applicability.	EBT achieves 99.1% accuracy on NSL-KDD.
Our paper	NSL-KDD	Combined CNN and channel attention	Focus on NSL-KDD limits generalizability. Future research needed.	Achieved 99.728% accuracy on NSL-KDD, surpassing previous methods.

3 Dataset

The NSL-KDD dataset, an improvement of the 1999 KDD Cup dataset, improves clarity and reduces redundancy for intrusion detection and classification models. The four subgroups' reduced

properties and examples aid network security research. The dataset comprises KDDTrain+ and KDDTest+ sets, with subcategories like KDDTrain+_20Percent and KDDTest-21 for evaluation.

Moreover, in the field of IDS and ML, the NSL-KDD dataset—a refined version of the original KDD99—has gained popularity. In contrast to its forerunner, NSL-KDD tackles some drawbacks such superfluous records and extraneous features, offering a more efficient and productive dataset for testing. Its higher quality makes more thorough analysis and model training possible, which enhances the efficacy of ML research and IDS. Researchers are beginning to see the advantages of NSL-KDD, and as a result, it is being widely adopted in both academia and industry, sometimes even surpassing the long-standing popularity of KDD99.

To update and streamline the data, the University of New Brunswick produced the NSL-KDD dataset [7,13,20,27]. Moreover, KDDTrain+_20Percent helps train and validate models with 20% of the training dataset. However, KDDTest-21, excluding record 21, allows controlled tests. The dataset supports supervised learning with 43 features, including 'attack' and 'level' labels for attack kind and intensity. The qualities of these properties classify them into four groups. For network security and intrusion detection, the NSL-KDD dataset enables comprehensive machine learning and data mining research.

The properties in the dataset can be classified into four distinct categories according to their inherent characteristics.

The dataset under examination has category, binary, discrete, and continuous features. Binary features (7, 12, 14, 20, 21, 22) describe attributes with two states, while categorical features (2, 3, 4, 42) reflect qualitative variables with distinct categories. Discrete features (8, 9, 15, 23–41, 43) are unique numeric variables, but continuous features (1, 5, 6, 10, 11, 13, 16, 17, 18, 19) can take any real value within a range.

The dataset's 'attack' label has 40 labels, categorizing attacks as revised, U2R, DoS, R2L, and probing. Each main class has subclasses with attack types. U2R elevates user privileges, DoS disrupts network traffic, R2L gains local access via remote systems, and Probe extracts information. Moreover, Table 2 shows the classifying attacks divided into five classes.

Table 2: Classifying attacks into five primary classes

Classes	DoS	R2L	Probe	U2R
Subclasses	apache2	ftp_write	ipsweep	buffer_overflow
	back	guess_passwd	mscan	loadmodule
	land	HTTP tunnel	nmap	Perl
	Neptune	imap	portsweep	ps
	mailbomb	multihop	saint	rootkit
	pod	named	satan	sqlattack
	processable	phf		xterm
	smurf	send email		
	teardrop	Snmpgetattack		
	udpstorm	spy		
	worm	snmpguess		
		warezclient		

(Continued)

Table 2 (continued)

Classes	DoS	R2L	Probe	U2R
		warezmaster xlock snoop		
Total	**11**	**15**	**6**	**7**

To summarize, the study utilizes a dataset that is extensive and varied, incorporating a broad spectrum of characteristics and forms of attack. The organized architecture of this system allows for the creation and assessment of machine learning models for detecting intrusions, which helps to enhance cybersecurity research. The comprehensive analysis of feature categories and assault classifications establishes a strong basis for the study and enables a meticulous examination of the dataset's complexities.

With a total of 39 types of attacks in addition to the 'normal' class, the total will be 40 subclasses.

Further, Table 3 displays the categories of datasets with records. The research report emphasizes datasets' usefulness in network threat prediction models. While the dataset is divided into training and test sets, subclass distribution differs greatly. It is useful for predicting primary classes and assessing network protocols, services, and flags. The dataset comprises category elements like protocol type, service, and flag and critical attributes like 'attack' for prediction models.

Table 3: Types of datasets with records

Dataset	Records					
	Total	Normal	DoS	Probe	U2R	R2L
KDDTrain+	12597 3	67343 (53%)	45927 (37%)	11656 (9.11%)	52 (0.04%)	995 (0.85%)
KDDTest+	22544	9711 (43%)	7458 (33%)	2421 (11%)	200 (0.9%)	2654 (12.1%)

Fig. 2 shows the relationship between attack kinds and network communication protocol flag values. Understanding each attack's communication protocols requires understanding the protocol type characteristic, which has three values: ICMP, TCP, and UDP.

Moreover, Fig. 3 shows the distribution of attack methods over each protocol's 11 flag values, helping identify patterns and potential malicious activity. It illuminates network attacks across communication channels.

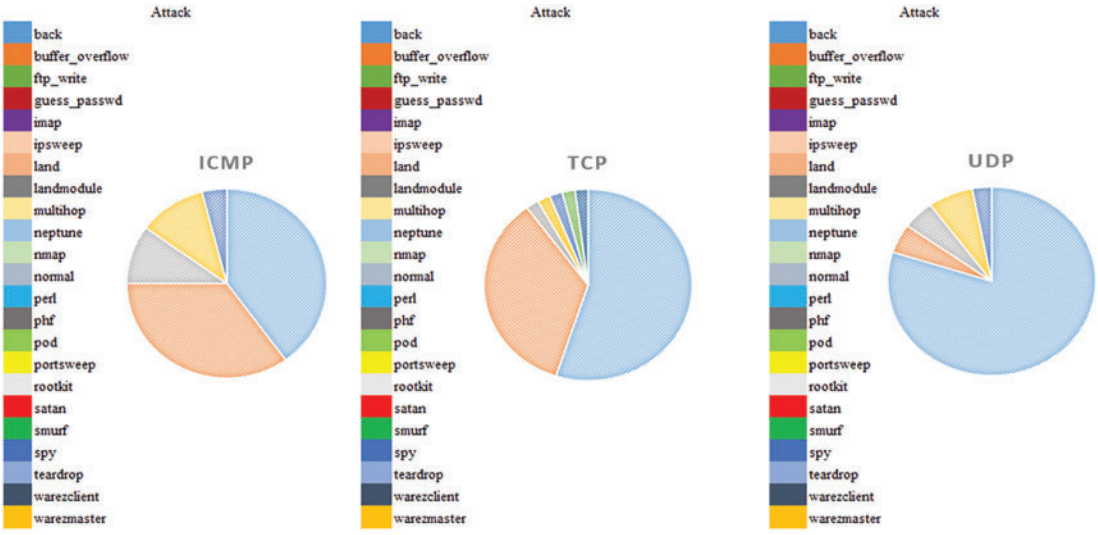

Figure 2: Types of attacks according to each protocol

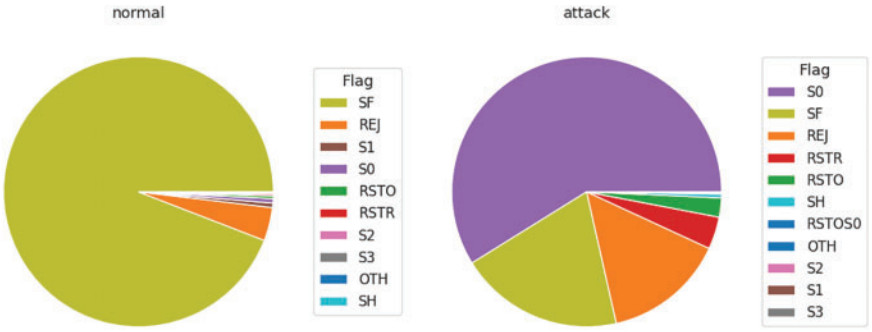

Figure 3: Flag types according to attack type

4 Methodology

The suggested research aims to use a CNN that combines channel attention to build a robust intrusion detection system (IDS). By using the NSL-KDD dataset, this effort seeks to increase intrusion detection accuracy and efficacy. The approach is divided into discrete phases that include data processing, feature engineering, and CNN implementation with a channel attention mechanism.

4.1 Data Processing and Feature Engineering

To begin, it is necessary to obtain the NSL-KDD dataset, which is a highly regarded benchmark dataset commonly employed for intrusion detection purposes [37,38]. The collection consists of a wide range of network traffic data, encompassing both regular and harmful activity.

i. Data Processing

In-depth data processing is done to confirm that the dataset is suitable for training a CNN. This includes handling abnormal data points, removing duplicate entries, and managing null values. The initial stage is to identify categorical traits. Next, categorical variables are converted into a numerical

format that neural networks can comprehend using the appropriate preprocessing techniques, such as one-hot encoding [38].

ii. Enhancing the Characteristics of the Data

Efficient feature engineering is essential for improving the performance of the intrusion detection system. The work concentrates on converting unprocessed data into a structure that captures fundamental patterns for categorization. This process entails the identification and categorization of certain characteristics, the transformation of tables into two-dimensional arrays, and the utilization of mathematical operations on these arrays for subsequent examination [38].

iii. Matrix Padding and Normalization

To ensure uniform input size for the CNN, matrix padding is used to accommodate the varying durations of network traffic sequences. This guarantees uniformity in the input dimensions, enabling smooth incorporation into the neural network structure. Furthermore, matrix normalization is utilized to rescale the feature values, enhancing the convergence and stability of the training process as shown in Fig. 4.

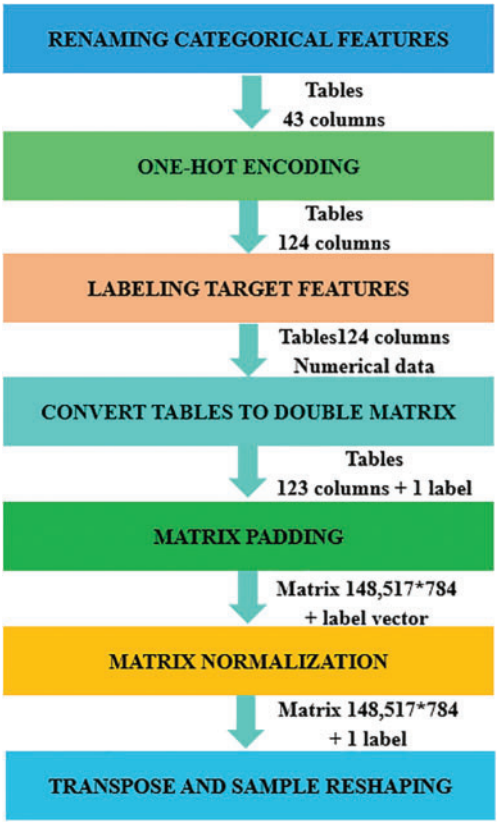

Figure 4: Data processing steps

iv. Rearrangement and Extraction of Samples

To optimize the utilization of the CNN architecture, the matrices are rearranged by transposing them to match the anticipated input format. This stage enhances the compatibility between the data representation and the convolutional layers of the network. Afterward, data samples are produced to

make a dataset that is evenly distributed, so resolving any problems related to imbalanced classes that could impact the model's effectiveness.

v. CNN Utilizing Channel Attention

The central component of the proposed Intrusion Detection System (IDS) is a CNN structure enhanced with channel attention methods. Channel attention boosts the model's capacity to concentrate on significant characteristics, hence enhancing classification precision. The CNN is trained using the preprocessed and engineered dataset, and hyperparameter tuning is performed to optimize the model's performance [38].

vi. Assessment and Verification

The ultimate stage entails assessing the CNN-based IDS using established performance indicators, including accuracy, precision, recall, and F1-score. Cross-validation techniques are used to assure the reliability of the model, and the suggested approach's effectiveness is validated by comparing it with existing intrusion detection approaches.

To summarize, the methodology described above presents a structured approach for creating a CNN Channel Attention Intrusion Detection System utilizing the NSL-KDD dataset. This approach focuses on data processing, feature engineering, and incorporating sophisticated neural network techniques.

4.2 Key Steps to Preprocess the Dataset and Train the CNN Model for Optimal Intrusion Detection Performance.

i. Categorical Feature Renaming: The NSL-KDD dataset comprises 43 features, with 4 being categorical. To streamline the categorization process, the objective feature 'attack' is converted from 40 distinct labels to the designated 5 major labels, namely Normal, DoS, U2R, R2L, and Probe. This stage optimizes the categorization task and guarantees that the model concentrates on the pertinent attack types.

ii. One-Hot Encoding: To format the dataset for use in the neural network model, the remaining 3 category characteristics undergo one-hot encoding. This procedure converts categorical data into numerical format by generating binary variables for every category. The numerical representation improves the model's capacity to acquire knowledge from the encoded characteristics, as shown in Fig. 5.

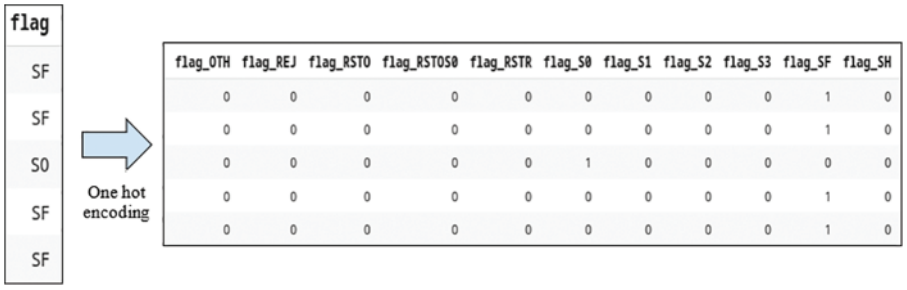

Figure 5: Example of encoding the feature fFlag' using one-hot encoding technique

iii. Conversion of Target Variable: The target variable 'attack' is transformed into a numerical format to simplify the model training process. The transformation gives the attack types

numerical labels: 0 represents normal, 1 represents denial of service (DoS), 2 represents Probe, 3 represents remote to local (R2L), and 4 represents user to root (U2R).

iv. ***Data Frame to Double Matrix Conversion:*** Since the dataset is initially in Pandas data frame format, it is necessary to convert it into a 2D array (matrix) to input the data into the neural network. The matrix format is compatible with the input requirements of the CNN model, allowing for smooth incorporation into the training process. Moreover, Fig. 6 shows the data frame as a 2D array.

```
array([[    0.,   491.,     0., ...,     1.,     0.,     0.],
       [    0.,   146.,     0., ...,     1.,     0.,     0.],
       [    0.,     0.,     0., ...,     0.,     0.,     1.],
       ...,
       [    0., 54540.,  8314., ...,     1.,     0.,     1.],
       [    0.,    42.,    42., ...,     1.,     0.,     0.],
       [    0.,     0.,     0., ...,     0.,     0.,     2.]]
```

Figure 6: Data frame in a 2D array (148517 × 124)

v. ***Input Reshaping for Neural Network Input:*** To prepare the input for the CNN model, which operates on images with dimensions (N, N), where N is a specified size, matrix padding is employed. This transformation enables each row of the matrix to be converted into a square shape of dimensions N2. For this study, two distinct N (12 and 28) values are chosen to generate two augmented datasets.

vi. ***Dataset Transformation:*** Utilizing matrix padding leads to generating two separate datasets with different resolutions. This approach is employed to assess the influence of input size on the performance of the CNN model. The prepared datasets are transformed for training and subsequent comparison of outcomes. Below, Fig. 7 shows the array of padding method.

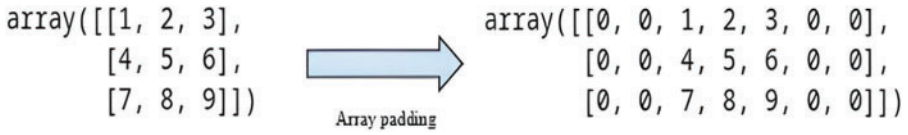

Figure 7: Example of array padding (Same padding method used)

vii. ***Matrix Normalization:*** Before inputting the data into the CNN model, it is subjected to matrix normalization to normalize the range of values. It is essential to take this step to avoid certain traits' dominance and ensure that the model is not biased towards particular attributes. The MinMaxScaler is selected as the normalization technique, which transforms each data point (X_i) based on the following formula: The outcome is a standardized dataset in which all values are confined to intervals between 0 and 1.

$$X_{norm} = \frac{X_i - min(X)}{max(X) - min(X)} \tag{1}$$

viii. ***Reshaping Example:*** After normalizing, the dataset is converted into square matrices, an essential process for preparing the data for input into a CNN. Two separate datasets are used, with initial dimensions of (148517 × 144) and (148517 × 784). The samples are further transformed into 2D square matrices, yielding two distinct types of samples: (12 × 12) and (28

× 28). The process of reshaping the data allows for the extraction of spatial features, which in turn enables the network to successfully identify patterns. Fig. 8 visually depicts several examples, demonstrating converting raw data into reshaped matrices.

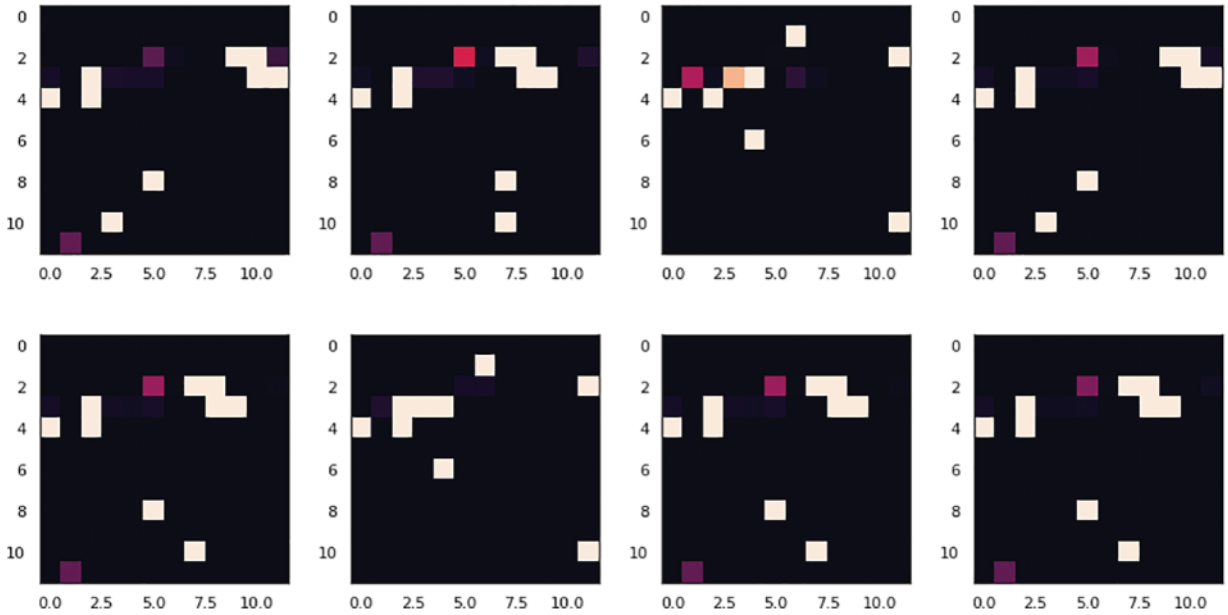

Figure 8: Sample arrays plotted as images

4.3 Efficient Channel Attention

The Efficient Channel Attention (ECA) mechanism has been recognized as a powerful attention mechanism for deep CNNs. It has shown impressive performance improvements while keeping the number of parameters relatively low. This study examines the ECA module, its superiority over alternative attention systems, and its application in the NSL-KDD dataset for a CNN-based intrusion detection system (IDS). Fig. 9 depicts the comparison of various attention modules.

i. Analyzing Attention Mechanisms in Convolutional Neural Networks

Attention processes in CNNs have a crucial impact on enhancing the representation of features and the model's overall performance. They facilitate the concentration of networks on informative regions and features while reducing irrelevant ones. Squeeze-and-excitation networks (SE-net) have been acknowledged for their efficacy; however, they entail heightened intricacy and computing requirements. The SE module often employs fully-connected layers and global average pooling (GAP) to capture non-linear cross-channel interactions. Nevertheless, reducing dimensionality in SE modules might lead to inefficiencies and unintended consequences in channel prediction [38–40].

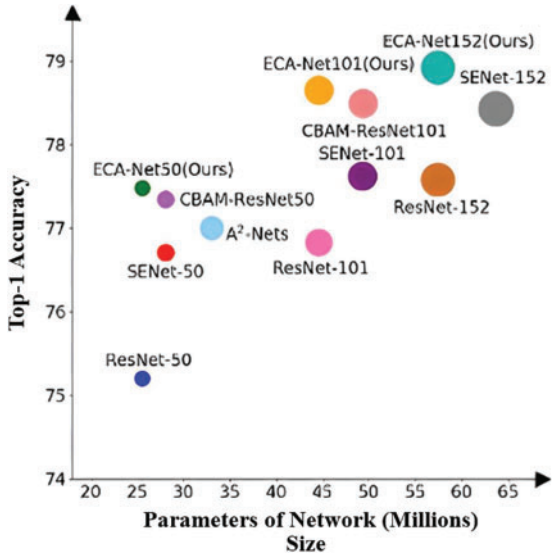

Figure 9: Comparison of various attention modules

ii. The Efficient Channel Attention (ECA) Module

The ECA module presents an innovative method for recording cross-channel interactions without utilizing dimensionality reduction, hence overcoming the constraints of SE modules. Unlike SE, ECA examines each channel and its k neighboring channels. This is achieved by employing a 1D convolution layer with a kernel size of k, efficiently capturing the local interactions among channels.

iii. Enhancing Efficiency in Interactions across Multiple Channels

The main benefit of ECA is its capacity to effectively capture interactions across different channels. Instead of depending on worldwide operations, ECA considers local neighborhoods, enabling a more intricate comprehension of feature relationships. This is accomplished by utilizing a 1D convolution process, which allows for the inclusion of local cross-channel interactions.

iv. Efficiency of Parameters

The ECA model demonstrates significant improvements in performance while utilizing a relatively limited set of parameters. This critical component guarantees that the model maintains a low weight and high computational efficiency. The ECA module's simplified design enhances its efficacy in diverse applications, such as intrusion detection.

v. Comparison with Other Attention Modules

To emphasize the benefits of ECA, it is crucial to do a comparative analysis with other attention mechanisms, such as SE and CBAM modules.

a) Challenges faced in the SE module

Although SE modules provide exceptional performance, using global average pooling and fully connected layers imposes significant computational overhead. The dimensionality reduction process, aimed at handling complexity, has exhibited inefficiency and unintended consequences in channel prediction. ECA addresses these difficulties by directly recording local cross-channel interactions, hence obviating the necessity for dimensionality reduction [39,40].

b) Comparison between ECA and CBAM

The efficacy of ECA is particularly evident when compared to the Convolutional Block Attention Module (CBAM). CBAM integrates channel and spatial attention mechanisms, augmenting computational complexity. On the other hand, ECA attains same or better results with a more straightforward structure, making it a more desirable option for applications with limited resources, such as intrusion detection.

vi. Utilization of the NSL-KDD Dataset for Intrusion Detection

Its practicality is demonstrated by the addition of ECA (Enhanced Channel Attention) to a CNN-based Intrusion Detection System (IDS) that uses the NSL-KDD dataset. The model's ability to distinguish between benign and malevolent network events depends on its ability to record cross-channel interactions. The intrusion detection system will remain extremely responsive and appropriate for deployment in various network circumstances thanks to the lightweight design of the ECA module.

Above, Fig. 10 shows the se module of the left-hand side and ECA Module on the right-hand side.

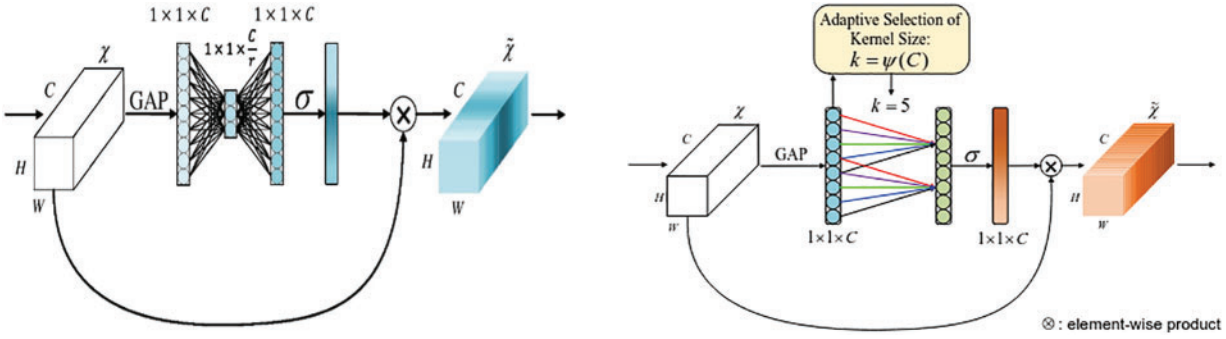

Figure 10: Diagram of SE-module (Left) *vs.* ECA-module (Right)

Moreover, the ECA mechanism is becoming increasingly popular in deep CNNs, especially in scenarios where there is a need for lightweight models that can deliver high performance. ECA surpasses conventional attention modules such as SE [41] and CBAM [41] by disregarding the necessity for dimensionality reduction and emphasizing local cross-channel interactions. Within the framework of an intrusion detection system that utilizes the NSL-KDD dataset, including ECA is proven to be an indispensable element. ECA significantly enhances the accuracy and efficiency of recognizing potential security threats. As technology progresses, attention methods such as ECA remain crucial in influencing the development of deep learning applications, guaranteeing strong and effective solutions in many fields.

4.4 Architectural Element

The NSL-KDD dataset is used in the design phase of a convolutional neural network (CNN) model for an intrusion detection system (IDS), and careful consideration of numerous architectural features is necessary to achieve maximum performance in detecting network intrusions. Within the CNN model, every layer is essential to extracting pertinent characteristics from the input data and effectively utilizing the ECA mechanism for improved attention allocation. Here, we explore the thinking behind the decisions we made on the design of each layer.

i. Input Layer

The NSL-KDD dataset provides the raw network traffic data to the CNN's input layer. This layer acts as the neural network's entry point for input processing. Since the input layer connects with the dataset directly and does not require any complicated changes, choosing it is simple [38,40].

ii. Convolutional Layers

Fundamental elements of CNNs, convolutional layers are in charge of applying convolutional operations to extract features. These layers seek to locate spatial patterns and structures in the network traffic data that can point to malicious activity in the context of intrusion detection [38–40]. The network can detect more abstract and complicated information by adding several convolutional layers, which improves its ability to distinguish between typical and aberrant network behavior.

iii. Activation Functions

Activation functions like as Rectified Linear Unit (ReLU) are used to make the network less linear after every convolutional operation. ReLU in particular facilitates faster computation of gradients, which helps ease the vanishing gradient problem and promotes convergence during training. ReLU is the preferred activation function due to its simplicity, efficiency, and capacity to promote sparse activation in the network.

iv. Pooling Layers

Pooling layers are used to downsample the feature maps produced by the convolutional layers. They are commonly implemented as max-pooling or average-pooling operations [38,40]. Because pooling layers lower the size of the feature maps while retaining the most crucial data, the translated model is more reliable and runs faster. Incorporating pooling layers facilitates the extraction of resilient features, encourages parameter sharing, and mitigates overfitting of the model.

v. Efficient Channel Attention (ECA) Mechanism

The CNN model can now be modified to adjust feature calibration according to the way each channel depends on the ECA mechanism. ECA makes it easier to selectively improve informative feature channels while suppressing noisy or irrelevant ones by explicitly modeling inter-channel connections. By dynamically altering the relevance of various feature channels, this attention method improves the CNN's overall performance in intrusion detection tasks and increases its discriminative power.

In conclusion, careful consideration of each layer's functionality and contribution to the system's overall performance is part of the CNN model for intrusion detection design process. The resulting CNN model can efficiently and accurately identify and classify network intrusions by utilizing convolutional layers for feature extraction, activation functions for introducing non-linearities, pooling layers for downsampling, and integrating the ECA mechanism for attention allocation.

4.5 Model Implementation

The CNN Channel Attention Intrusion Detection System was implemented using the NSL-KDD dataset. This was achieved using the ECA-Net module, which is crucial in enhancing the model's ability to detect intricate characteristics in the input data. The module was specifically designed as a component within the TensorFlow 2.0 framework, utilizing pre-existing layers like *Conv2D* and *GlobalMaxPooling2D* to provide a customized attention mechanism [38,39].

The ECA-Net module is inspired by the ECA mechanism, which presents a new method for recalibrating channel-wise features in CNNs as shown in Fig. 11. This method effectively extracts pertinent information from several sources, allowing the network to prioritize the most informative characteristics while suppressing irrelevant ones. The purpose of including the ECA-Net module was to enhance the intrusion detection system's capability to identify subtle trends in network traffic that may indicate potential security risks.

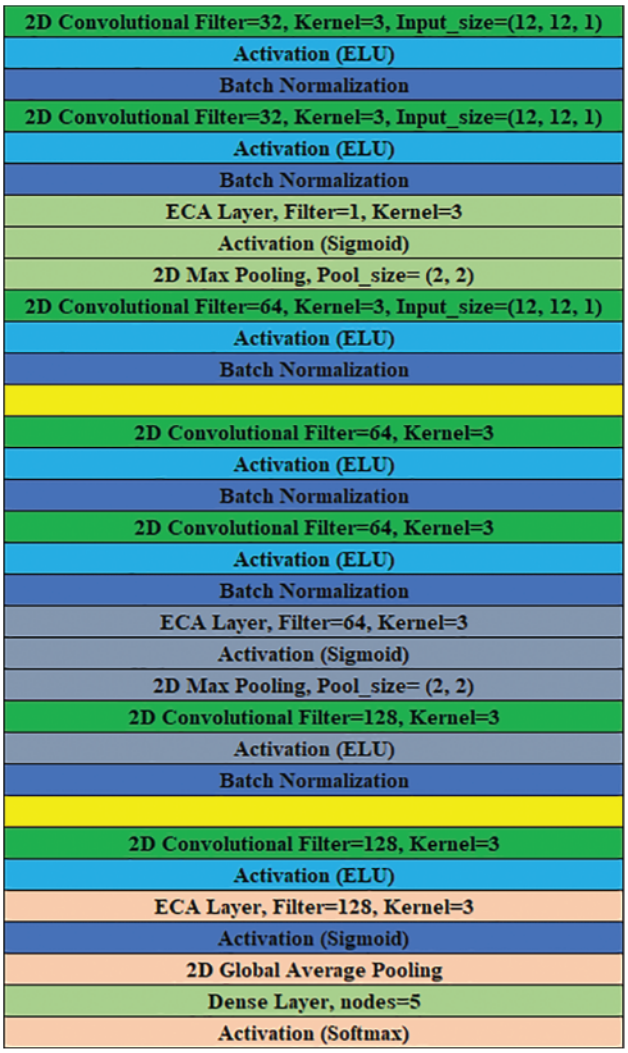

Figure 11: Convolutional neural network with ECA layers architecture

After the ECA-Net module was created, the remaining parts of the network were developed according to the architectural blueprint shown in Fig. 10. This architectural design functioned as a blueprint, determining the organization and interconnections of succeeding layers to guarantee a unified and efficient neural network. The network has 288,357 parameters, designed specifically to handle input samples with dimensions of 12×12 and 28×28.

The Adam optimizer was utilized with a learning rate 0.001 to enhance the network's parameters and promote efficient learning. Utilizing the sparse categorical cross-entropy loss function throughout

the training process was highly beneficial in quantifying the dissimilarity between the predicted and real labels. Having established these settings, the model was trained for 100 epochs, with each epoch involving the iteration over a dataset divided into 90% training data, 9% validation data, and 1% testing data.

The training method was conducted on Google's Colab Research platform, utilizing the computational capabilities of NVIDIA T4 Tensor Core GPUs. Utilizing this advanced computing system allowed for effective parallel processing, accelerating the training iterations and promoting the convergence of the model. Using GPUs is highly beneficial in deep learning applications because it expedites matrix computations used in neural network training, resulting in substantial savings in training time.

The CNN Channel Attention Intrusion Detection System was developed by first creating the ECA-Net module. This module is a customized attention mechanism that improves the model's ability to extract features. Subsequently, the network was meticulously constructed, the parameters were fine-tuned, and the training process was thoughtfully strategized. The outcome yielded a robust intrusion detection system capable of analyzing and categorizing network traffic patterns to identify potential security risks.

4.6 Model Evaluation

Using a CNN Channel Attention model, the study highlights the need of channel-wise feature selection to improve intrusion detection accuracy. Moreover, by employing the channel attention strategy, the model can focus on relevant features, which improves its overall performance. In addition, recall, F1-score, accuracy, and precision are used to assess a model's performance.

- The harmonic means of precision and recall, or F1-score, provides a fair assessment of model performance.
$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2}$$
- It takes recall, which gauges the model's accuracy in identifying real positives, to find every instance of intrusion.
$$Recall = \frac{TP}{TP + FN} \tag{3}$$
- The accuracy score indicates that the model can distinguish between undesired and typical network activities, demonstrating its dual detection capabilities.
$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \tag{4}$$
- Precision is a parameter used to assess the model's ability to reduce false positives, which measures the percentage of correctly detected positive cases out of all anticipated positive cases.
$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Using the NSL-KDD dataset, the research offers a comprehensive analysis of a CNN Channel Attention Intrusion Detection System. A comprehensive evaluation using measures like accuracy, precision, recall, F1-score, and AUC reveals that the proposed CNN model performs better than the current models. The study opens the door for future advancements in designing robust and effective network security solutions and offers important insights into intrusion detection.

5 Results

The assessment of the suggested system entails a meticulous examination of the training and validation accuracy and the loss performance throughout numerous epochs.

An important consideration for evaluating the model's ability to learn and generalize is the training and validation accuracy performance curve. The model's performance on the training set is reflected in the training accuracy, which evaluates the model's capacity to predict the labels for the training data. By evaluating the model's performance on a separate validation set that was not used for training, validation accuracy gauges the model's ability to generalize to new data. Effective learning and generalization are demonstrated when the performance curve's slope increases for training and validation accuracy. A persistent discrepancy between the training and validation sets' accuracy could be a sign of overfitting, emphasizing the need for regularization techniques.

A useful tool for understanding the convergence and optimization of the model during training is the training and validation loss performance curve. The main objective of training is to reduce the loss, a metric that measures the difference between expected and actual values. A decrease in the model's training and validation losses indicates an improvement in its ability to produce accurate predictions. However, a growing difference between the training and validation losses may indicate overfitting, or the model becoming overly adapted to the training data.

Fig. 12, which shows the relationship between epochs and performance, is essential to understanding how the model improves over time. The number of epochs, or complete iterations over the entire training dataset, is represented by the x-axis. The pertinent accuracy and loss values are shown on the y-axis. When the model finds patterns in the data, the loss decreases, but the training and validation phases' accuracy increases initially. However, to determine the optimal number of epochs, it is crucial to closely observe the point at which the model begins to overfit or converge.

Ultimately, a comprehensive understanding of the CNN Channel Attention IDS—developed using the NSL-KDD dataset—is provided by the performance curves and the relationship between epochs and performance. These visualizations help assess the model's ability to learn, extrapolate to new data, and optimize while it is being trained. By thoroughly examining these graphs, researchers and industry professionals can enhance the model's efficacy in intrusion detection scenarios by optimizing hyperparameters.

5.1 Confusion Matrix

The Confusion Matrix is an essential tool when assessing the CNN Channel Attention Intrusion Detection System's performance using the NSL-KDD dataset. It provides a comprehensive examination of the model's predictions and actual outcomes.

True negative (TN) indicates accurate identifications of non-intrusive occurrences, whereas true positive (TP) refers to situations where the system correctly detects intrusions. A false positive (FP) happens when the model erroneously identifies normal activities as intrusions, while a false negative (FN) indicates cases where true intrusions are mistakenly categorized as normal.

Based on the NSL-KDD dataset, the confusion matrix is shown in Fig. 13 and shows the true labels *vs.* predicted labels of the CNN Channel Attention Intrusion Detection System. To facilitate the evaluation of model performance, this visual representation offers a concise summary of true positives, true negatives, false positives, and false negatives.

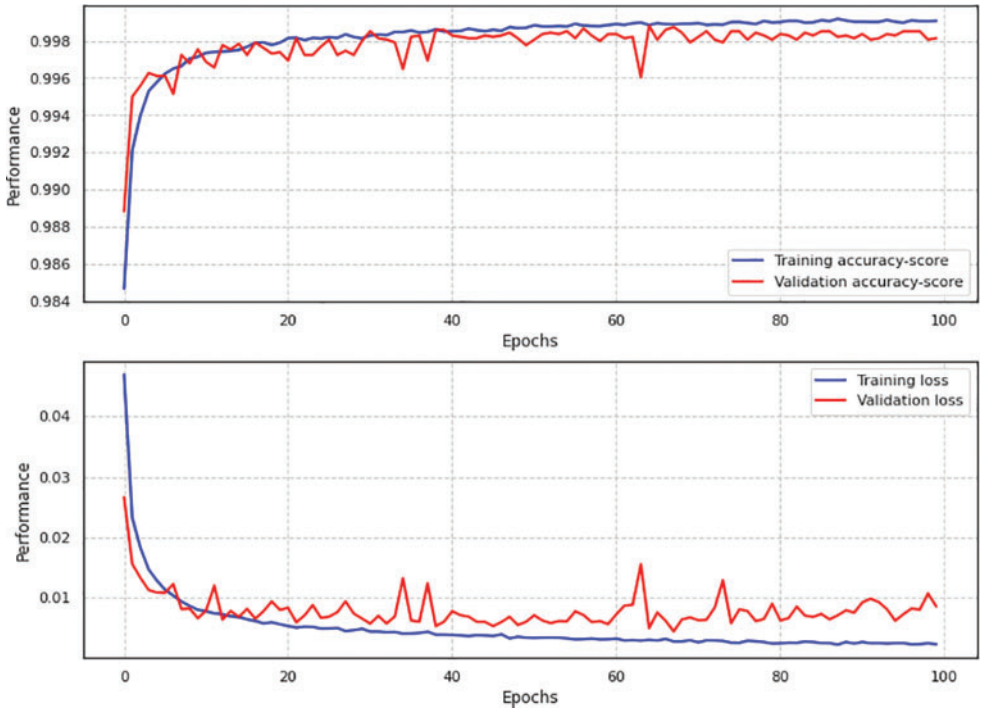

Figure 12: Training and validation accuracy and loss performance

5.2 Comparative Analysis

In Table 4, the proposed CNN Channel Attention Intrusion Detection System outperforms existing models with perfect scores on all tests using the NSL-KDD dataset. With flawless scores of 0.99 in accuracy, precision, recall, and F1-score, our model stands out from earlier models that have demonstrated varied degrees of performance. In contrast, reference [7] obtains a good 0.843 accuracy but does poorly in terms of recall and F1-score. In a similar vein, references [10] and [13] show passable accuracy but neither recall nor precision. Reference [20] exhibits subpar accuracy performance, pointing to a large false positive rate. Although it lacks recall data, reference [23] displays unbalanced results with respectable precision and F1-score. Although reference [27] obtains excellent ratings, our proposed model performs better, especially when it comes to precision. Our model is preferable because it can attain high accuracy with similarly excellent precision, recall, and F1-score. In intrusion detection systems, where minimizing false alarms and improving detection accuracy are critical, this all-encompassing performance is essential. Our suggested model shows that it is effective in improving intrusion detection and network security due to its outstanding performance on all metrics.

6 Discussion

A noteworthy development in network security is the CNN Channel Attention Intrusion Detection System, which uses the NSL-KDD dataset. Incorporating the ECA-Net module was a key factor in enhancing the model's ability to detect complex features in the input data. In the age of developing cyber dangers, there is an increasing need for effective intrusion detection systems, and this creative solution aims to meet that requirement.

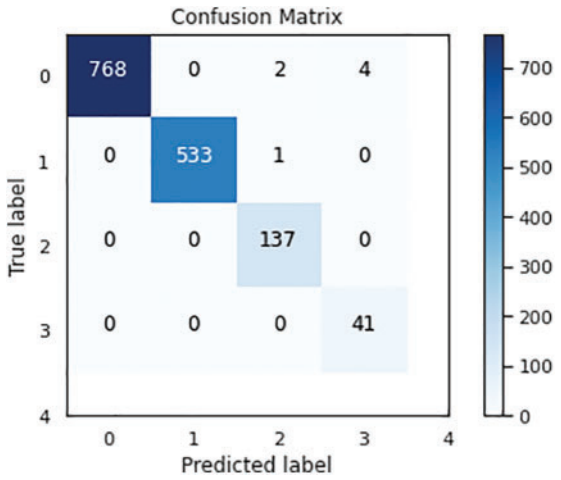

Figure 13: Confusion matrix

Table 4: Performance comparison of our proposed model with other references

Models	Methods	Accuracy	Precision	Recall	F1-score
[7]	Hybrid Auto-encoder with CNN	0.843	0.933	0.781	0.85
[10]	MCNN	0.694	0.84	0.69	0.74
[13]	ANN	0.78	0.966	0.6205	0.7557
[20]	CNN	0.79	0.234	0.686	–
[23]	RBM (Boltzman Machine)	0.7323	0.6233	–	0.753
[27]	ANN	0.975	0.99	0.967	0.957
Our proposed model	**CNN with channel attention**	**0.99**	**0.99**	**0.99**	**0.99**

i. Importance of Deep CNNs' ECA Mechanism

Our model's ECA mechanism has become well-known in deep convolutional neural networks (CNNs). It excels especially when striking a compromise between high performance and model complexity is essential. The model is superior at identifying subtle patterns in network traffic due to the ECA mechanism's capacity to capture channel-wise dependencies, making it perfect for intrusion detection applications as shown in Fig. 14.

ii. Comparing Our Model with Others

A thorough comparison between the CNN Channel Attention Intrusion Detection System and other models that are currently in the literature is part of our research. Performance measures were used in the evaluation on the NSL-KDD dataset. The outcomes showed that our model outperformed

every other model examined in earlier references. This accomplishment demonstrates how well the suggested strategy provides enhanced intrusion detection capabilities.

Figure 14: Effective intrusion detection system

iii. Accuracy and Performance Metrics

The precision, accuracy, recall, and F1-score metrics assessed the model's performance. With an outstanding accuracy rate of 99.728%, our CNN Channel Attention Intrusion Detection System exhibits remarkable accuracy. This outcome shows how well the model can discriminate between typical network behavior and invasive activity.

iv. Strongness and Broadness

Our model's resilience and ability to generalize to various network circumstances is one of its noteworthy features. Using the ECA-Net module, the CNN with channel attention produced reliable and efficient results under various network circumstances. This robustness is essential in real-world applications where network settings might be dynamic and prone to change.

v. Adjustability for Compact Models

The need for high-performing, lightweight models in the intrusion detection is always increasing. Our CNN Channel Attention Intrusion Detection System meets this requirement, which includes the ECA-Net module. The model demonstrates that cutting-edge performance can be attained without compromising computing economy. The practicality and scalability of our suggested solution are improved by its adaptation to lightweight architectures.

vi. Consequences for Real-World Intrusion Monitoring

Our model's outstanding results on the NSL-KDD dataset have encouraging ramifications for actual intrusion detection applications. Preserving confidential data and preserving the integrity of networked systems require the capacity to precisely detect and categorize network breaches. According to our research, the CNN Channel Attention Intrusion Detection System may be useful for improving different companies' security postures.

vii. Restrictions and Upcoming Developments

Although our study offers a strong intrusion detection system, it must be acknowledged that it has limitations. Subsequent investigations may examine possible improvements aimed at mitigating these constraints and enhancing the model's functionality. Furthermore, further research is necessary to confirm the generalizability and efficacy of the CNN Channel Attention Intrusion Detection System in a variety of scenarios before applying it to additional datasets and real-world network environments.

Furthermore, the NSL-KDD dataset exhibits significant improvements compared to the KDD99 dataset, making it a more advantageous choice for modern IDS and MLR efforts. The training and evaluation benefits from its refined and balanced nature, which effectively solves the inherent shortcomings of its predecessor. NSL-KDD employs careful curation to create a more accurate representation of real-world network traffic, hence reducing biases and inconsistencies that are inherent in KDD99. Furthermore, the dataset covers a wider range of attack types and scenarios, enhancing its appropriateness for thorough intrusion detection investigations. NSL-KDD's growing popularity among researchers highlights its effectiveness in producing strong and widely applicable models. Looking towards the future, it is anticipated that NSL-KDD will continue to be the favored option for researchers exploring the complexities of machine learning and network intrusion detection in this field. The results of our study have important implications for network security, highlighting the capacity of our methodology to make a substantial contribution to the detection and reduction of cyber risks. It is recommended to pursue additional research and improvement in order to enhance the capabilities of intrusion detection systems and strengthen the resilience of networked systems against the ever-changing problems posed by cybersecurity.

7 Conclusion

In short, this research study used a convolutional neural network (CNN) with a channel attention mechanism to provide a novel approach for intrusion detection systems. The NSL-KDD dataset was used to test and apply the suggested methods. Promising outcomes have been observed when the ECA mechanism is added to deep convolutional neural networks (CNNs), particularly in scenarios where very effective and efficient models are required. After extensive testing, we obtained an astounding accuracy of 99.728% for our suggested model, which outperformed all other models examined in this research.

This research demonstrates the efficacy of CNN with channel attention mechanisms through a comparative examination. Significantly, our model achieved superior performance compared to several existing models, such as the hybrid auto-encoder with CNN, MCNN, ANN, adaptive algorithm, CNN, RBM (Boltzman machine), ANN, and ensemble learning. The significant superiority of our model over these benchmarks emphasizes its resilience in identifying intrusions in network traffic, showcasing its potential for practical use. Moreover, the importance of intrusion detection systems in protecting networks from hostile activities cannot be overemphasized. The growing intricacy and refinement of cyber threats necessitates the implementation of innovative and effective intrusion detection technologies. Our suggested CNN with channel attention model demonstrates exceptional accuracy, making it a strong contender for use in real-world situations. This would greatly improve cybersecurity measures.

Although this research has been successful, there are still opportunities for further investigation and enhancement. Subsequent research should prioritize improving the capacity of the suggested model to be applied to many datasets other than the NSL-KDD dataset, thereby increasing its

generalizability. Assessing the model's performance across various network topologies and traffic patterns would yield useful insights into its flexibility and resilience in diverse situations.

Additionally, it is important to examine the scalability of the Convolutional Neural Network (CNN) when incorporating channel attention mechanisms, especially in network infrastructures of significant scale. It is essential to assess the model's performance as network traffic volume grows to determine its suitability for enterprise-level networks and critical infrastructure.

Furthermore, integrating real-time functionalities into the intrusion detection system is a crucial area for future research, alongside scalability. Real-time detection and response to breaches are essential for minimizing the potential harm caused by cyber-attacks. Investigating methods to decrease latency and enhance response times will enhance the feasibility and efficiency of the suggested paradigm in dynamic network situations.

Further research can be conducted to investigate the application of transfer learning techniques toto improve the model's performance in situations where labeled data are scarce. Transfer learning is the utilization of knowledge acquired from one domain to enhance the performance of a model in a related but distinct domain. This is especially significant in cybersecurity, where acquiring labeled data can be difficult.

This study represents a notable advancement in intrusion detection systems by introducing a CNN with a channel attention mechanism. This model demonstrates exceptional accuracy when applied to the NSL-KDD dataset. Our model's encouraging results and the suggested topics for future research establish it as a vital contribution to the ongoing efforts to strengthen network security against increasing cyber threats. The continuous development of intrusion detection field leads to valuable knowledge that will enable the creation of more sophisticated and adaptable cybersecurity solutions.

7.1 Future Work

Using the NSL-KDD dataset, this paper offers a unique method for intrusion detection using a convolutional neural network (CNN) and channel attention mechanisms. Even though the performance is encouraging, there are a number of directions this research might go in the future to be further explored and improved.

i. **Exploration of Alternative Attention Mechanisms:** Although channel attention has been demonstrated to enhance feature representations, investigating varying attention mechanisms such as spatial attention or multi-head attention may provide additional insights into how to enhance the model's ability to discriminate between objects. More resilient intrusion detection systems may result from research on the interactions and complementarities of various attention mechanisms.
ii. **Integration of Advanced Architectures:** Opportunities to enhance model performance arise from ongoing developments in deep learning architectures. More complicated patterns in network traffic data may be captured by integrating more sophisticated designs, such as transformer-based networks or hybrid models that combine CNNs with recurrent structures. This would improve the system's capacity to identify complex intrusions.
iii. **Feature Engineering and Selection:** The effectiveness of machine learning models is greatly dependent on feature engineering. More investigation into feature engineering techniques tailored to network intrusion detection, such using domain expertise or extracting domain-specific features, may result in more valuable representations and more precise detection.
iv. **Data Augmentation and Imbalanced Data Handling:** Resolving dataset imbalances and improving the model's generalizability across various intrusion classes continue to be significant

problems. Techniques like data augmentation, the synthetic minority oversampling technique (SMOTE), or ensemble learning approaches that are made to function with irregular data could be used to solve these issues. Additionally, this would increase the overall reliability of the intrusion detection system.

v. **Adversarial Robustness:** As security threats change, it is more important than ever to make sure intrusion detection systems are resilient to adversarial attacks. The system may become more dependable in practical deployment scenarios if methods such as adversarial training, defensive distillation, or input sanitization are investigated to strengthen the model's resistance to modifications made by adversaries.

vi. **Real-Time Implementation and Deployment:** Practical factors like computational efficiency and real-time processing requirements must be addressed in order to move the established model into real-world deployment situations. For practical applicability, methods for hardware acceleration, quantization, or model optimization that are designed to deploy CNN-based intrusion detection systems in resource-constrained contexts must be investigated.

vii. **Evaluation on Diverse Datasets:** The NSL-KDD dataset serves as a reference for studies on intrusion detection. A more thorough understanding of the suggested method's effectiveness in actual use would be obtained by testing it on various datasets that depict various network settings and attack scenarios.

To sum up, the above-mentioned future work offers a number of viable avenues for improving the suggested CNN Channel Attention Intrusion Detection System. Our goal in pursuing these paths is to enhance the system's functionality, resilience, and suitability for practical cybersecurity situations.

Acknowledgement: The authors would like to thank Princess Nourah bint Abdulrahman University for funding this project through the Researchers Supporting Project (PNURSP2023R319) and Prince Sultan University for covering the article processing charges (APC) associated with this publication. Special acknowledgement to Automated Systems & Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia. Also, the authors wish to acknowledge the editor and anonymous reviewers for their insightful comments, which have improved the quality of this publication.

Funding Statement: The authors would like to thank Princess Nourah bint Abdulrahman University for funding this project through the Researchers Supporting Project (PNURSP2023R319) and this research was funded by the Prince Sultan University, Riyadh, Saudi Arabia.

Author Contributions: Study conception and design: Fatma S. Alrayes, M. Zakariah, S. U. Amin; data collection: M. Zakariah, S. U. Amin, Z. I. Khan; analysis and interpretation of results: Fatma S. Alrayes, S. U. Amin, Z. I. Khan; draft manuscript preparation: M. Zakariah, J, S. Alqurni, S. U. Amin. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Dataset is available on reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Mirlekar, K. P. Kanojia, and B. Chourasia, "A stacked CNN-BiLSTM model with majority technique for detecting the intrusions in network," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 55, pp. 152–162, 2024.

[2] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach," *Expert. Syst. Appl.*, vol. 238, no. 1, pp. 121751, Mar. 2024. doi: 10.1016/j.eswa.2023.121751.

[3] V. Hnamte and J. Hussain, "DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system," *Telemat. Inform. Rep.*, vol. 10, pp. 100053, Jun. 2023. doi: 10.1016/j.teler.2023.100053.

[4] K. Pramilarani and P. V. Kumari, "Cost based random forest classifier for intrusion detection system in internet of things," *Appl. Soft Comput.*, vol. 151, pp. 111125, Jan. 2024. doi: 10.1016/j.asoc.2023.111125.

[5] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. C. Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alex. Eng. J.*, vol. 61, no. 12, pp. 9395–9409, Dec. 2022. doi: 10.1016/j.aej.2022.02.063.

[6] M. H. Nasir, J. Arshad, and M. M. Khan, "Collaborative device-level botnet detection for internet of things," *Comput. Secur.*, vol. 129, pp. 103172, Jun. 2023. doi: 10.1016/j.cose.2023.103172.

[7] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, "A survey of CNN-based network intrusion detection," *Appl. Sci.*, vol. 12, no. 16, pp. 8162, Aug. 2022. doi: 10.3390/app12168162.

[8] K. Sakthi and P. N. Kumar, "A novel attention-based feature learning and optimal deep learning approach for network intrusion detection," *J. Intell. Fuzzy Syst.*, vol. 45, no. 3, pp. 5123–5140, Aug. 2023. doi: 10.3233/JIFS-231758.

[9] Z. Wang and F. A. Ghaleb, "An attention-based convolutional neural network for intrusion detection model," *IEEE Access*, vol. 11, pp. 43116–43127, 2023. doi: 10.1109/ACCESS.2023.3271408.

[10] I. Al-Turaiki and N. Altwaijry, "A convolutional neural network for improved anomaly-based network intrusion detection," *Big Data*, vol. 9, no. 3, pp. 233–252, Jun. 2021. doi: 10.1089/big.2020.0263.

[11] J. Du, K. Yang, Y. Hu, and L. Jiang, "NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning," *IEEE Access*, vol. 11, pp. 24808–24821, Mar. 2023. doi: 10.1109/ACCESS.2023.3254915.

[12] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A deep learning model for network intrusion detection with imbalanced data," *Electronics*, vol. 11, no. 6, pp. 898, Mar. 2022. doi: 10.3390/electronics11060898.

[13] S. Sapre, P. Ahmadi, and K. Islam, "A robust comparison of the KDDCup99 and NSL-KDD IoT network intrusion detection datasets through various machine learning algorithms," 2019. doi: 10.48550/arXiv.1912.13204, 2019.

[14] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Comput. Electr. Eng*, vol. 99, Feb. 2022. doi: 10.1016/j.compeleceng.2022.107810.

[15] H. E. Namiq, A. D. Salman, and A. M. Dinar, "An improved intrusion detection system based on convolutional neural networks (CNN) algorithm," *Res. Sq.*, vol. 2, no. 2, pp. 843, 2023. doi: 10.21203/rs.3.rs-3096808/v1.

[16] P. Iyer, T. Jadhav, A. Pillai and Samundiswary, "Analysis of modern intrusion detection algorithms and developing a smart IDS," in *2021 Int. Conf. Intell. Technol. (CONIT)*, IEEE, 2021, pp. 1–7. doi: 10.1109/CONIT51480.2021.9498519.

[17] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82512–82521, 2019. doi: 10.1109/ACCESS.2019.2923640.

[18] S. Karthic and S. M. Kumar, "Hybrid optimized deep neural network with enhanced conditional random field based intrusion detection on wireless sensor network," *Neural Process. Lett.*, vol. 55, no. 1, pp. 459–479, Jun. 2022. doi: 10.1007/s11063-022-10892-9.

[19] A. K. Samha, N. Malik, D. Sharma, S. Kavitha, and P. Dutta, "Intrusion detection system using hybrid convolutional neural network," *Mob. Netw. Appl.*, vol. 92, no. 4, pp. 106301, Aug. 2023. doi: 10.1007/s11036-023-02223-6.

[20] M. E. Magdy, A. M. Matter, S. Hussin, D. Hassan, and S. Elsaid, "A comparative study of intrusion detection systems applied to NSL-KDD dataset," *Egypt. Int. J. Eng. Sci. Technol.*, vol. 43, no. 2, pp. 88–98, Sep. 2023. doi: 10.21608/eijest.2022.137441.1156.

[21] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. 9th EAI Int. Conf. Bio-Insp. Inf. Commun. Technol.*, ACM, 2016. doi: 10.4108/eai.3-12-2015.2262516.

[22] M. Vishwakarma and N. Kesswani, "DIDS: A deep neural network based real-time intrusion detection system for IoT," *Decis. Anal. J.*, vol. 5, no. 1, pp. 100142, Nov. 2022. doi: 10.1016/j.dajour.2022.100142.

[23] Y. Imamverdiyev and F. Abdullayeva, "Deep learning method for denial of service attack detection based on restricted boltzmann machine," *Big Data*, vol. 6, no. 2, pp. 159–169, 2018. doi: 10.1089/big.2018.0023.

[24] S. Wang, W. Xu, and Y. Liu, "Res-TranBiLSTM: An intelligent approach for intrusion detection in the internet of things," *Comput. Netw.*, vol. 235, pp. 109982, 2023. doi: 10.1016/j.comnet.2023.109982.

[25] O. F. Isife, K. Okokpujie, I. P. Okokpujie, R. E. Subair, A. A. Vincent and M. E. Awomoyi, "Development of a malicious network traffic intrusion detection system using deep learning," *Int. J. Saf. Secur. Eng.*, vol. 13, no. 4, pp. 587–595, 2023. doi: 10.18280/ijsse.130401.

[26] S. Alzughaibi and S. El Khediri, "A Cloud intrusion detection systems based on DNN using backpropagation and PSO on the CSE-CIC-IDS2018 dataset," *Appl. Sci.*, vol. 13, no. 4, pp. 2276, 2023. doi: 10.3390/app13042276.

[27] M. Zakariah, S. A. AlQahtani, A. M. Alawwad, and A. A. Alotaibi, "Intrusion detection system with customized machine learning techniques for NSL-KDD dataset," *Comput. Mater. Contin.*, vol. 77, no. 3, pp. 4025–4054, 2023. doi: 10.32604/cmc.2023.043752.

[28] O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, "EIDM: Deep learning model for IoT intrusion detection systems," *J. Supercomput.*, vol. 79, no. 12, pp. 13241–13261, 2023. doi: 10.1007/s11227-023-05197-0.

[29] W. L. Al-Yaseen and A. K. Idrees, "MuDeLA: Multi-level deep learning approach for intrusion detection systems," *Int. J. Comput. Appl.*, vol. 45, no. 12, pp. 755–763, 2023. doi: 10.1080/1206212X.2023.2275084.

[30] A. V. Turukmane and R. Devendiran, "M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning," *Comput. Secur.*, vol. 137, no. 1, pp. 103587, 2024. doi: 10.1016/j.cose.2023.103587.

[31] Q. Abu Al-Haija and A. Al-Badawi, "Attack-aware IoT network traffic routing leveraging ensemble learning," *Sensors*, vol. 22, no. 1, pp. 241, 2021. doi: 10.3390/s22010241.

[32] A. Kayyidavazhiyil, "Intrusion detection using enhanced genetic sine swarm algorithm based deep meta-heuristic ANN classifier on UNSW-NB15 and NSL-KDD dataset," *J. Intell. Fuzzy Syst.*, vol. 45, no. 6, pp. 10243–10265, 2023. doi: 10.3233/JIFS-224283.

[33] S. Mohamed and R. Ejbali, "Deep SARSA-based reinforcement learning approach for anomaly network intrusion detection system," *Int. J. Inf. Secur.*, vol. 22, no. 1, pp. 235–247, 2023. doi: 10.1007/s10207-022-00634-2.

[34] S. Kanumalli, K. Lavanya, A. Rajeswari, P. Samyuktha, and M. Tejaswi, "A scalable network intrusion detection system using Bi-LSTM and CNN," in *2023 Third Int. Conf. Artif. Intell. Smart Energy (ICAIS)*, IEEE, 2023, pp. 1–6. doi: 10.1109/ICAIS56108.2023.10073719.

[35] R. Harini, N. Maheswari, S. Ganapathy, and M. Sivagami, "An effective technique for detecting minority attacks in NIDS using deep learning and sampling approach," *Alex. Eng. J.*, vol. 78, no. 1, pp. 469–482, 2023. doi: 10.1016/j.aej.2023.07.063.

[36] A. T. Azar, E. Shehab, A. M. Mattar, I. A. Hameed, and S. A. Elsaid, "Deep learning based hybrid intrusion detection systems to protect satellite networks," *J. Netw. Syst. Manage.*, vol. 31, no. 4, pp. 82, Oct. 2023. doi: 10.1007/s10922-023-09767-8.

[37] R. Qaddoura, A. M. Al-Zoubi, H. Faris, and I. Almomani, "A multi-layer classification approach for intrusion detection in IoT networks based on deep learning," *Sensors*, vol. 21, no. 9, pp. 2987, Apr. 2021. doi: 10.3390/s21092987.

[38] Y. Ding and Y. Zhai, "Intrusion detection system for NSL-KDD dataset using convolutional neural networks," in *Proc. 2018 2nd Int. Conf. Comput. Sci. Artif. Intell.*, New York, NY, USA, ACM, Dec. 2018, pp. 81–85. doi: 10.1145/3297156.3297230.

[39] C. E. Asry, S. Douzi, and B. E. Ouahidi, "A deep learning model for intrusion detection with imbalanced dataset," in *Advances in Intelligent System and Smart Technologies*, 2024, vol. 11, no. 6, pp. 261–271. doi: 10.1007/978-3-031-47672-3_26.

[40] P. Dixit and S. Silakari, "Application of deep learning techniques in cyber-attack detection," in *Soft Computing: Theories and Applications*. Singapore: Springer, 2022, vol. 1380, no. 1, pp. 229–241. Accessed: Apr. 28, 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-16-1740-9_20

[41] Z. Zhao, F. Wang, S. Chen, H. Wang, and G. Cheng, "Deep object segmentation and classification networks for building damage detection using the xBD dataset," *Int. J. Digit. Earth*, vol. 17, no. 1, pp. 1–20, Dec. 2024. doi: 10.1080/17538947.2024.2302577.