aurum

AURUM MÜHENDİSLİK SİSTEMLERİ VE MİMARLIK DERGİSİ

*AURUM JOURNAL OF ENGINEERING SYSTEMS AND ARCHITECTURE*

Cilt 7, Sayı 2 | Kış 2023
Volume 7, No 2 | Winter 2023

# A NEW SOFTWARE DEFINED NETWORKS (SDN) IN IOTS BASED DEEP LEARNING TECHNIQUES

Osamah AL-HWAIDI[1], Hasan Hüseyin BALIK[2]
[1] Electrical and Computer Engineering, Altınbaş University, İstanbul, Türkiye
, 203720699@ogr.altinbas.edu.tr
(https://orcid.org/0000-0001-7962-4366)
[2] Electrical and Computer Engineering, Yildiz Technical University, İstanbul, Türkiye,
hasanbalik@gmail.com
(https://orcid.org/0000-0003-3022-100X)

Received: 22.02.2023 Research Article
Accepted: 11.05.2023 pp.165-185
Published: 31.12.2023
*Corresponding author DOI:10.XXXX/ajesa.2023.XXXXX

## Abstract

In this study, a new software defined networks (SDN) in IoTs based on deep learning techniques was implemented using various types of classifiers such as DNN, CNN, GRU, LSTM RNN and SDN Ryu controller. The system was able to handle high-dimensional and complex data by using NSL-KDD dataset, and was able to detect unknown intrusions that traditional methods may miss. The effectiveness of the proposed model was evaluated by accuracy, precision, recall, F-score, and confusion matrix. Python 3.10 has been used to implementation our system. The proposed system was able to achieve good performance, however, the system's efficacy will be determined by the kind of the data feed and the scale of the issue that is attemped to address. This study highlights the potential of DL-based NIDS with SDN and IoT to detect network intrusions, but also highlights the need for continuous monitoring and updating to ensure that the system remains effective.

**Keywords:** Deep Learning, SDN, IDS, IOT.

## IOTS TABANLI DERİN ÖĞRENME TEKNİKLERİNDE YENİ BİR YAZILIM TANIMLI AĞLAR (SDN)

## Özet

Bu çalışmada, DNN, CNN, GRU, LSTM ve RNN ve SDN Ryu denetleyici gibi çeşitli sınıflandırıcılar kullanılarak IoT tabanlı derin öğrenme tekniklerinde yeni bir yazılım tanımlı ağlar (SDN) gerçekleştirilmiştir. Sistem, NSL-KDD veri setini kullanarak yüksek boyutlu ve karmaşık verileri işleyebildi ve geleneksel yöntemlerin gözden kaçırabileceği bilinmeyen izinsiz girişleri tespit edebildi. Modellerin etkinliği, doğruluk, kesinlik, hatırlama, F-puanı ve karışıklık matrisi ile değerlendirildi. Python 3.10 sistemi uygulamak için kullanılmıştır. Önerilen sistem iyi bir performans elde etmeyi başarmış, ancak sistemin etkinliği, onu besleyen verilerin türüne ve ele almaya çalışılab sorunun ölçeğine göre belirlenmektedir. Bu çalışma, ağ izinsiz girişlerini algılamak için SDN ve IoT ile DL tabanlı NIDS'nin potansiyelini vurgulamaktadır, ancak aynı zamanda sistemin etkili kalmasını sağlamak için sürekli izleme ve güncelleme ihtiyacını da vurgulamaktadır.

**Anahtar Kelimeler:** Derin Öğrenme, SDN, IDS, IOT.

## 1. Introduction

The traditional network model is being challenged by the emergence of Software-Defined Networking (SDN) with deep learning and Internet of Things (IoT) technology, SDN simplifies network management and provides greater flexibility and scalability, making it ideal for applications such as network intrusion detection systems (Ambika, 2021). Deep learning algorithms can be used to provide better security, detect malicious traffic more accurately, and identify new threats faster, with the help of IoT devices, SDN networks can also be monitored in real-time to detect any suspicious activities or anomalies (Saritha, 2022). This brings us to the question of how traditional networks compare to SDN networks with deep learning and IoT when it comes to network intrusion detection systems (Hendrawan, 2019). With the advent of new technologies such as deep learning, IOT and network intrusion detection systems, traditional networks are no longer sufficient for modern businesses, software defined networking (SDN) is a more efficient and cost-effective way to manage networks, it allows for greater flexibility in terms of design, configuration and deployment (Li, 2019). Additionally, SDN can be combined with deep learning and IOT to provide enhanced security and improved performance, Moreover, SDN can also be used to detect any malicious activities on the network using a Network Intrusion Detection System (NIDS), this will help organizations protect their data from cyber threats and ensure that their networks are secure (Varghese,2021). Traditional networks are becoming outdated as the demand for more efficient and secure networks increases, with the advent of Software Defined Networks (SDN) and deep learning, we can now create a network that is more secure, reliable, and cost-effective (Ong, 2014). Furthermore, with the integration of IOT devices into this network, we can also detect any potential intrusions in real-time (Raj, 2018). SDN networks with deep learning and IOT have revolutionized the way in which we create networks, these networks are much faster and more secure than traditional ones, this makes them ideal for organizations that require high levels of security (Sutton, 2021). Intrusion detection systems (IDS) are becoming increasingly important for businesses, as cyber-attacks and malicious activities become more sophisticated, by combining Software Defined Networking (SDN) and deep learning with Internet of Things (IoT), companies can create an intrusion detection system that is both reliable and secure (Sutton, 2021).

SDN allows for the automation of network management, while deep learning enables the detection of complex patterns and anomalies in data traffic, this makes it possible to detect malicious activities quickly and accurately, Furthermore, by incorporating IoT devices into the system, companies can monitor a wide range of devices simultaneously (Zwane, 2019). Using an intrusion detection system with SDN and deep learning along with IOT can help businesses protect their networks from cyber-attacks and other malicious activities, companies can also benefit from improved efficiency and cost savings due to automated network management processes (Usman, 2020).

This research issue is motivated by the rising need of protecting IoT networks from attack. We can no longer imagine our lives without the Internet of Things, and the number of linked devices continues to increase. The growth of IoT devices, however, has increased the risk for hacking and network penetration. However, traditional NIDS may not be able to identify new or developing threats in IoT settings since they depend on preset signatures (Alhowaide, 2021). The suggested system intends to use deep learning approaches to deal with high-dimensional and complicated data, both of which are prevalent in IoT settings. The suggested NIDS uses deep learning and Software Defined Networking (SDN) to detect attacks that more conventional approaches could miss. The suggested NIDS also incorporates software-defined networking (SDN) in an effort to provide a more flexible and malleable solution for protecting IoT settings. Since SDN allows the partitioning of the control plane from the data plane, network assets may be managed in a more dynamic and centralized manner. The suggested research challenge also incorporates the implementation of the system in a real-world setting, as well as the assessment of its performance using several metrics including accuracy, precision, recall, and F-score (Panigrahi,2022). This will aid in determining the efficacy of the proposed NIDS in discovering new and undiscovered network intrusions in IoT settings using SDN.

These are the specific, realizable goals that make up the overall aim:

1. To create state-of-the-art procedures for NIDS and the best applicable approaches.
2. The second goal is to use NIDS datasets to test and compare different Deep learning strategies.
3. Create and implement an intrusion detection system (NIDS) prototype for IoT networks.
4. Assessing the efficiency of the suggested NIDS.

This study investigated how the most up-to-date cybersecurity and networking developments might be used to identify and counteract the most common forms of assault.

To begin, Software Defined Networking (SDN) has been identified as one of the few developments that offers several benefits to IOT Networks. Accordingly, this study lays the groundwork for future research into SDN as a security enhancer solution and encourages more research and investigation of SDN's possibilities in IOT Networks. In this research, a NIDS model was developed that takes full use of SDN to sample and prepare flow data quickly and easily. By developing a prototype for a NIDS in (SDN) environment, the researcher of this research shown that SDN may be an effective complement to intrusion detection systems (IDSs).

Further, Deep Learning (DL) is one of the rapidly developing topics, with a lot of research being done in the pursuit of efficient methods for detecting network intrusions. However, most NIDS-related research just analyzes the effectiveness of various Machine Learning algorithms on datasets, rather than showing how they may be put to practical use. An ineffective NIDS cannot be guaranteed by comparing DL techniques using datasets gathered online, which is a major downside. Therefore, a prototype of the suggested model was developed using the Design Science approach. This approach provides a useful model for doing research that has practical and theoretical implications. This methodology may be used by other researchers in the area to create, deploy, and test their own suggested DL-based IDSs in addition to analyzing datasets.

In this contribution, we used SDN in conjunction with an effective DL model to develop a smart NIDS that can collect information from IOT network devices. Such data is then processed and analyzed to detect invasive behaviors, and it is equipped to take immediate, hands-off corrective action.

## 2. Related Studies

Here, we go into the current state of knowledge and discuss how it relates to the larger discussion around IoTs security. An appropriate NIDS framework may be discovered by conducting a comprehensive analysis of state-of-the-art technology. This research also aids in the identification of possible paradigms that might handle some of the communication and network security problems in IoTs. The first step of the chosen technique, Design Science Research, is a thorough examination of the issue and a search for gaps in the current literature on the subject. Consequently, this section of the study investigates the many technological options available for use in realizing the study's objective.

A method for detecting DDoS attacks in consumer IoT networks using classification algorithms is proposed by (Doshi, 2018). Figure 1 depicts the detection process and every IP address, port number, packet size, and time stamp are recorded, as well as the destination and source IP addresses and ports. Finally, data packets are sorted by their source IP addresses. As with the data coming from each individual device, the packet stream is broken up into discrete intervals that don't overlap. Packet size, IP Address Range, and Protocols are stateless characteristics that are obtained; stateful attributes include Total Bandwidth and Overall Number of Unique Target IP Addresses. In this experiment, three IoT devices were used to simulate actual web traffic. The DDoS attacks were generated by simulating common techniques used in these types of attacks, such as TCP SYN flooding, UDP flood, and HTTP GET flood. A total of five supervised machine learning algorithms (KNN, SVM with a linear kernel, Decision Tree, Random Forest, and FCNN) were tested and compared. Depending on the techniques used, the researchers reported an F1 score of 0.927 to 0.999.

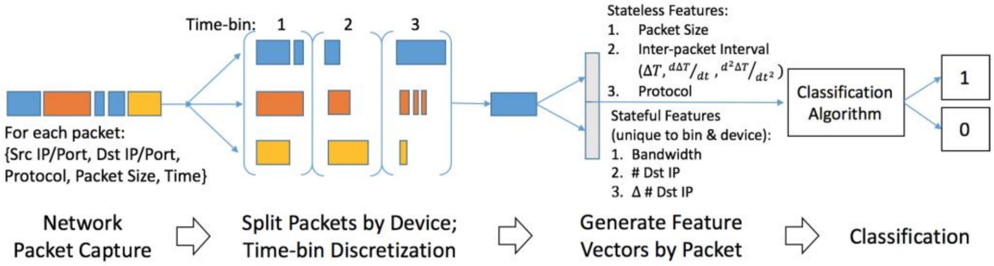

Time-bin: 1 2 3
For each packet:
{Src IP/Port, Dst IP/Port,
Protocol, Packet Size, Time}
Stateless Features:
1. Packet Size
2. Inter-packet Interval
$(\Delta T, {}^{d\Delta T}/_{dt}, {}^{d^2\Delta T}/_{dt^2})$
3. Protocol
Stateful Features
(unique to bin & device):
1. Bandwidth
2. # Dst IP
3. Δ # Dst IP
Classification Algorithm
1
0
Network Packet Capture
Split Packets by Device; Time-bin Discretization
Generate Feature Vectors by Packet
Classification

**Figure 1.** DDoS attack in IoT (Doshi, 2018).

For detecting attacks on the Internet of Things (Diro, 2018), present a distributed deep learning-based system. The attack detection models are trained and stored at the fog devices (switches and routers, gateways, hubs, etc.) in the periphery of the dispersed network. Each fog node's trained model shares its parameters with the others thanks to a coordinating master node. Specifically, they suggest using a fully convolutional neural network (FCNN) with three hidden layers (150, 120, and 50 neurons per layer, respectively) to identify malicious or benign network activity. Instead of analyzing data from actual IoT networks, they use the NSL-KDD data - set to evaluate the efficacy of their strategy. The model is able to identify attacks with a 99% efficiency and a false positive rate (FPR) of 0.85%.

Using a network-based approach (Hodo, 2016) offer a technique of anomaly detection for the IoT called N-BAIOT. In order to identify IoT botnets, they suggest using stacked autoencoders. The packet's sending and receiving hosts and protocols are analyzed in real time to create a behavioral snapshot. Over the course of many time periods, 115 statistics were calculated. The statistics include (1) all traffic that started at the same IP address, (2) all traffic that started at the same MAC address and IP address, (3) all traffic that started at the source IP and ended at the destination IP, and (4) all traffic that started at the source IP and ended at the destination TCP/UDP socket. Simply put, we only have the average and standard deviation of the data size and the IAT. Time ranges of 200 MS, 250 MS, 15 s, 20 s, and 1 minute are considered. It's important to remember that without the anomaly detector being set up on the local network, it's impossible to calculate such data. When training an autoencoder, only one instance is used across all devices. The testing step necessitates awareness of the source of the traffic. The experimental assessment makes use of nine Internet of Things gadgets. When the MIRAI and BASHLITE viruses are used in a controlled lab setting, they produce malicious traffic. Autoencoders use a 4-layer encoding and 4-layer decoders. The authors assert a 0.0070 FPR and a 100% attack detection accuracy.

An overview of ML and DM approaches in cybersecurity literature is offered in (Pajouh, 2016). The authors claimed that it was difficult to give a single suggestion for each work because of the fecundity and complexity of these strategies, and that thus, the most successful strategy for cyber apps has not been determined. What matters most is the kind of assault the system was built to stop. The authors went on to claim that there are other parameters that should be taken into account when evaluating the efficacy of ML/DM approaches, including the accuracy, complexity, classification speed, and interpretability of the final result. The significance of data sets for ML/DM in cyber intrusion detection was also stressed in their research. IDS benefit from access to system and kernel-level data for efficient anomaly or abuse detection, as described by (Pajouh, 2016); if at all practicable, data set should be supplemented with OS Kernel-level data. However, they limited their research to works that had already been done to evaluate ML/applicability DMs in the field of intrusion detection. The current research goes further that previous work25 by suggesting an IDS approach that makes use of ML and then assessing the system's efficacy.

(Meidan, 2018) developed a cross-layer paradigm to aid in the resolution of network security challenges in IOT networks. The framework's emphasis on efficiency and automation makes it a great fit for IOT networks. A key feature of the authors' suggested approach is its emphasis on coordinating security services across all tiers of communication. It was motivated by the discovery that processes at higher levels may be made safer or optimized by using data gathered from lower-level security services, such as authentication and IDS (Meidan, 2018). For instance, an interconnected cross-layer security mechanism may receive real-time threat profiles from authentication and IDS components running at the application layer. When these findings are compiled, they may be communicated to the lower levels to enhance their functionality and reliability. However, this approach raises the bar for inter-node connectivity and increases the difficulty and internal processing inside a node. The authors suggested that IDS, carrier frequency, and distributed verification are only some of the security services that may be incorporated using their platform.

According to (Buczak, 2015) SDN networks provide a useful framework for enforcing data-centric security standards across several layers of mobile communication networks. They maintained that rigorous experimental examination and validation of suggested technological ideas prior to implementation in real government systems was an essential part of building advanced security solutions. They suggested an OpenFlow-based testbed for verifying SDN security features. Using this technology, you may implement data-centric security regulations in addition to SDN layer protection methods. The study's findings backed up their method's capacity to verify simulation and analytical forecasts.

## 3. Methodology

Intrusion detection systems (IDS) have become increasingly important in today's connected world, as they are used to detect and prevent unauthorized access to networks and systems. One approach to NIDS is to use deep learning (DL) techniques in conjunction with (SDN) and the (IoTs) to create a more robust and accurate system. Deep learning systems can monitor network data for anomalies that can suggest a cyberattack, while SDN can be used to dynamically respond to detected threats by redirecting traffic or isolating compromised devices. IoT devices, such as cameras and sensors, can also be integrated into the system to provide additional data for analysis and improve the overall effectiveness of the IDS.

### 3.1. Proposed Approach

It is important to note that classical machine learning approaches have disadvantages when it comes to identifying and preventing cyber-attacks, and that as threat and penetration strategies and tools continue to develop, the implementation of these techniques may become less efficacious. Furthermore, machine learning approaches often require huge amounts of data, and this presents a problem when trying to build an effective and convenient (NIDS). In our approach we use several deep learning classifiers, each trained on 41 characteristics from the NSL-KDD dataset, in conjunction with SDN and IoTs technologies to improve the likelihood of detecting a wide range of intrusions. Our goal is to construct a more trustworthy and effective NIDS by learning the classifiers on the best correlation characteristics and primarily evaluating them on the accuracy of the matching measurement. Several criteria are used to assess the efficacy of our method, and the classifiers are comparing to one another and to those of previous studies. The IoTs devices could also provide information to the NIDS system for further evaluation and utilize it as an extra feature in the training dataset, also we employed five classifiers (GRU, DNN, LSTM, CNN, RNN) connect with SDN controller (Ryu) and IoTs devices to implementation our method. To further expedite training and get optimal results in creating a useful NIDS classifier, we used numericalize and normalize the data. In addition, we used a feature selection technique to choose the most relevant and useful information, which not only strengthens the system's resistance to cyber-attack but also prevents the training algorithm from overlooking crucial details. We believe our method has the potential to greatly enhance the detection and prevention of cyber-attacks by integrating the use of various classifiers, feature selection, numericalization, and normalization. Therefore, the model of our suggested procedure is shown in figure 2.

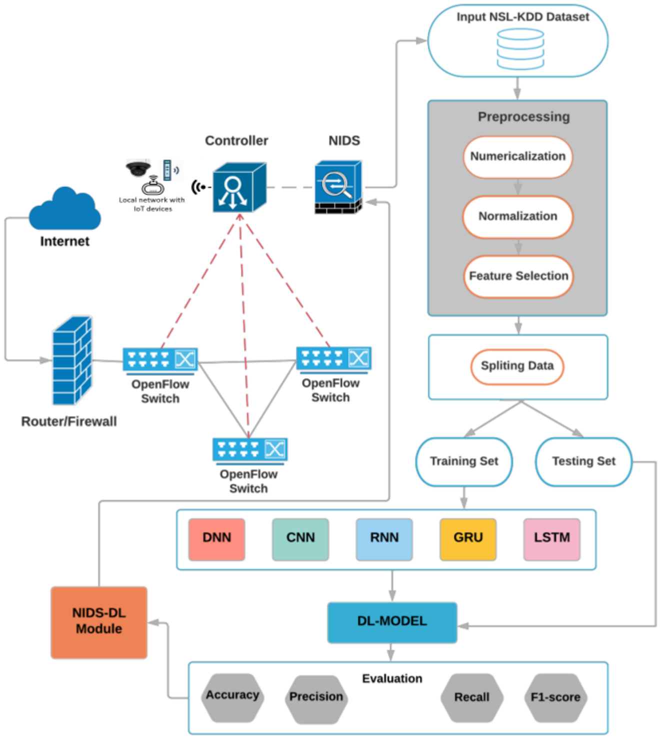


**Figure 2.** Model of our proposed approach.

The technology is effective because it makes use of deep learning methods to identify malicious network activity. It starts by inputting the NSL-KDD dataset, which is a widely used dataset for NIDS researches. The data then goes through a preprocessing step, which includes numericalization, normalization, and feature selection. Numericalization converts categorical data into numerical data, making data simpler for the model to process. Normalization scales the characteristics of the numerical data to a specific range, which helps to speed up the training process. Feature selection is implemented to select the most important attributes from a dataset that will be used in the training of the classifiers. Once the preprocessing step is complete, the data is divided into training and testing sets. The training set is used to train the classifiers, which include (GRU, DNN, LSTM, CNN, RNN). These classifiers are part of a deep learning model, which is trained to detect various types of network attacks. After the classifiers are trained, the testing set is inputted into the DL model to evaluate its performance. The evaluation process includes metrics such as F1-score, Precision, Accuracy, Recall, to measure the performance of the model and the results of the evaluation are then used to enhance the proposed approach. The proposed approach is then integrated into (NIDS) which is integrated with (SDN) controller and connected to the NSL-KDD dataset from other side. The SDN controller includes OpenFlow switches that are integrated with IoT devices. The OpenFlow switches are connected with a firewall, which generates an more layer of security for the network. Once the program is executed, it simulates the monitoring of the network for any suspicious activity, using the trained classifiers to detect and classify any potential attacks. If an attack is discovered, the system raises an alarm and takes necessary actions based on the type of attack, such as blocking the malicious traffic or sending an alert to the network administrator. In short, the system uses deep learning techniques to detect various types of network attacks, it uses SDN and IoT to improve efficiency and accuracy of the system and to make it more reliable and efficient against attacks, it continuously monitors the network, and takes action to prevent or mitigate any detected attacks.

### 3.2. Dl Model Classifier

Deep Learning Model classifier is a type of artificial intelligence that uses deep learning algorithms to classify data into different categories, it is used in applications such as image recognition, text classification, IDS and natural language processing (Shin, 2019). Deep Learning Model classifier with NIDS is a powerful tool to identify and classify malicious activities on a network, it uses deep learning algorithms to detect anomalies in the network traffic and identify suspicious activities, this technology can be used to detect malicious activities like DDoS attacks, malware, and phishing attacks, the Deep Learning Model classifier with NIDS can also be used for anomaly detection, intrusion detection, and threat intelligence, with its advanced capabilities, it can help organizations protect their networks from cyber threats (Kiran, 2022). In our approach we used 5 classifiers:

**Convolutional Neural Network (CNN):** CNN are a type of deep learning algorithms that are gaining popularity for their potential use in intrusion detection systems (IDS). CNNs have the ability to detect patterns in data and can be used to identify malicious activity. By using CNNs, IDS can detect intrusions more accurately and quickly than traditional methods, in addition, CNNs can also be used to identify new threats that have not been seen before, this makes them an ideal tool for security professionals who need to stay ahead of the curve when it comes to detecting malicious activity (Parkar, 2021).

**Deep Neural Network (DNN):** DNN are a sort of ML algorithm whose structure and operation are patterned after the human mind, DNN consist of numerous layers of linked processing and transmitting nodes or "neurons". DNN have shown potential in the realm of IDS, IDS are used to detect and respond to unauthorized access or malicious activity on a computer network (Alshahrani, 2022).

**Recurrent Neural Network (RNN):** RNN are a type of neural network that are designed to effectively handle sequential data, such as network traffic logs. RNNs are able to keep track of historical information and make decisions based on the context of previous data. This makes them well-suited for intrusion detection systems (IDS), where the goal is to identify patterns and anomalies in network traffic that may indicate an intrusion (Boné, 2011).

**Gated Recurrent Unit (GRU):** GRU is a powerful tool for intrusion detection systems (IDS), it provides an effective way to detect malicious activity in networks by recognizing patterns and anomalies in traffic. The GRU is a type of recurrent neural network (RNN) that uses gates to control the flow of information between layers, this makes it well suited for analyzing sequences of data, such as network traffic. By leveraging the GRU's ability to learn from past data, IDS can quickly identify suspicious behavior and alert administrators before malicious activity can cause damage. With its ability to

process large amounts of data quickly, the GRU is becoming an increasingly popular choice for intrusion detection systems (Salem, 2022).

**Long Short-Term Memory LSTM:** LSTM networks are a type of RNN structure that is designed to effectively handle sequential data, such as network traffic logs. LSTMs are able to keep track of historical information and make decisions based on the context of previous data, making them well-suited for intrusion detection systems (IDS), where the goal is to identify patterns and anomalies in network traffic that may indicate an intrusion (Dai, 2021).

**SDN Ryu CONTROLLER:** SDN is a network design in which the control plane, which makes choices on how data is transmitted via the network, is isolated from the data plane, which physically transmits the data. In an SDN architecture, a controller, such as Ryu, acts as the centralized decision-maker for the network (Nam, 2018).

### 3.3. Dataset

The NSL-KDD dataset is a widely used dataset for NIDS research that was created by modifying the KDD Cup 1999 dataset. The original KDD Cup dataset had several issues, such as the presence of redundant and irrelevant features, and a lack of diversity in attack types. The NSL-KDD dataset was created to address these issues, and is widely used in research on machine learning-based IDS in software-defined networks (SDN). The NSL-KDD dataset contains a total of around 25,000 instances, with 22 different types of attacks and 41 features. The features include information such as network protocol type, service, and flag. The dataset is split into two groups: a training set and a test set. The training set contains around 12,000 instances, and the test set contains around 13,000 instances (Meena, 2017). The dataset is balanced, with an equal number of normal instances and attack instances. In using this dataset for intrusion detection system research, it is common to preprocess the dataset to remove irrelevant features and reduce the dimensionality of the data. Various ML algorithms such as decision trees, random forests, and neural networks can be trained on the preprocessed data to classify instances in the test set as normal or attack. The performance of these models is typically evaluated using metrics such as accuracy, precision, recall, and F1-score.

~~3.4.~~

### ~~3.5.~~3.4. Preprocessing Of Data

Data preprocessing is an important step in using the NSL-KDD dataset for NIDS research. It involves cleaning, transforming, and normalizing the data to make it suitable for training machine learning models. Common preprocessing techniques that are applied to the NSL-KDD dataset include:

1. Feature Selection: This involves removing irrelevant features that do not contribute to the classification task, this can be done using techniques such as mutual information, chi-squared test, or correlation-based feature selection (Rahim, 2022).
2. Feature Extraction: This involves reducing the dimensionality of the data by combining or transforming the features. Common techniques include principal component analysis (PCA), linear discriminant analysis (LDA), and independent component analysis (ICA) (Sharma, 2019).
3. Data Normalization: This involves scaling the data to a specific range, usually between 0 and 1, to ensure that the features are on the same scale. This can improve the performance of some machine learning algorithms (Fong, 2021).
4. Numericalization is another preprocessing step that can be applied to the NSL-KDD dataset when using it for intrusion detection system research. It involves converting categorical features in the dataset into numerical values (Chen, 2012).

5. Outlier Detection: This involves identifying and removing instances in the dataset that are considered outliers, or instances that are significantly different from the majority of the data. This can be done using techniques such as the Z-score (Shou, 2018).
6. Data Balancing: This involves adjusting the number of instances of different classes in the dataset to ensure that the dataset is balanced (Wuisan, 2022).

### ~~3.6.~~3.5. Splitting Of Data

Data splitting is an important step in the ML process. It involves dividing the available dataset into two or more subsets, which are then used for different purposes. In the case of the NSL-KDD dataset, the data is split into a testing subset (25%) and a training subset (75%). The testing subset is used to evaluate the performance of a model, while the training subset is used to train the model. The goal of data splitting is to ensure that the model is not overfitting to the training data. Overfitting occurs when a model is too complex and is able to memorize the training data, but is not able to generalize to new, unseen data. By using a separate testing subset, the model's performance can be evaluated on data that it has not seen before. This allows for an assessment of the model's ability to generalize to new situations, which is an important aspect of the model's performance. Another goal of data splitting is to have representative sample of the entire data set in each subset, this is achieved by using different data splitting techniques such as random sampling, stratified sampling and others. Additionally, the data splitting should be done in a way that reduces the chances of data leakage between the training and testing sets. This is important as it can lead to an over-optimistic assessment of the model's performance. In summary, data splitting is an important step in the machine learning process that helps to ensure that the model is not overfitting to the training data, and that it can generalize to new, unseen data.

## 4. Performance Measures

To develop a dependable Network Intrusion Detection System (NIDS) and apply it effectively, it is crucial to achieve high performance when evaluated according to various metrics, particularly accuracy. One common way to evaluate the performance of a NIDS is to use various metrics based on the confusion matrix, such as accuracy, recall, precision, and F1-score. These metrics are calculated using parameters specific to the confusion matrix, such as True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

### 4.1. Results Of The Experiment

he objective of our methodology is to achieve optimal performance across multiple metrics. To accomplish this, we employed the Python 3.10 programming language in conjunction with TensorFlow and Keras libraries. Additionally, we utilized NumPy and Pandas for preprocessing tasks with 500 epochs. The hardware configuration used in this approach consisted of Intel Core i9-9900K processor, 64GB of RAM, an NVIDIA 8GB graphics card, and a 2TB SSD. This configuration was chosen to ensure that the necessary computational power was available for the task at hand.

### 4.2. Results Of The Proposed Models

Based on our data, it seems that our proposed models are very effective at the job it was designed to do. The models seem to be producing accurate predictions and accurately identifying a significant fraction of positive instances based on its high accuracy, precision, recall, and F1-score values. The models are very accurate, with a score between 0.9906 to 0.9892 indicating that more than 98% of the samples, the predictions are right. This means that the models are not overfitting to the training data and can successfully generalize to fresh data. With a high precision score between 0.9895 to 0.9881, the models have a more 98.8% success rate when making positive case predictions. As a result, the models are not producing a large number of false positive results. A recall score between 0.9921 to 0.9908 indicates that more 99% of true positives are being properly identified by the models. It’s reassuring to know that the models aren’t likely to be ignoring any genuine positives. The F1-score between 0.9908 to 0.9894 is a good value since it is the harmonic mean of precision and recall, which is a useful parameter for comparing models. The models have a fair balance between precision and recall if this number is larger than either precision or recall alone. Our proposed Models performance measures are shown in figure 3.

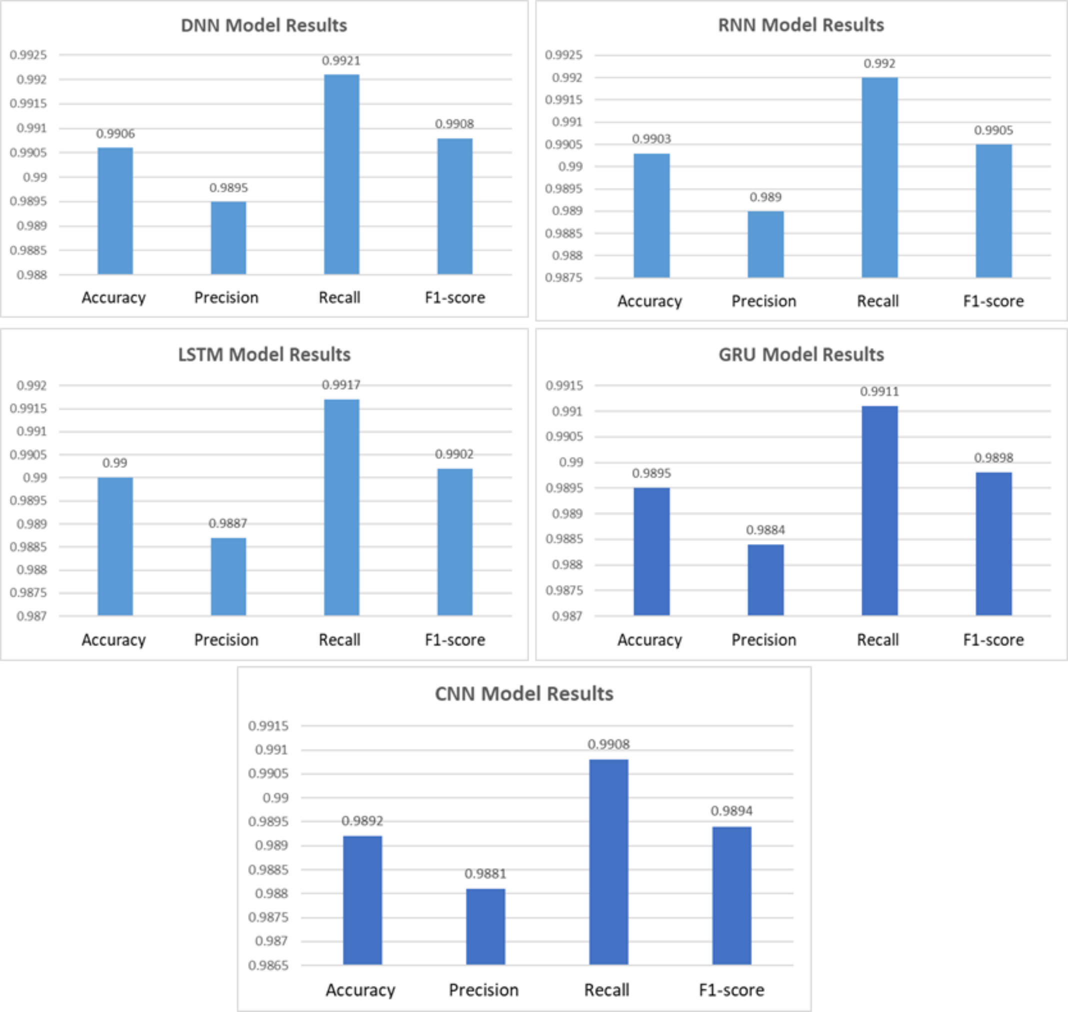
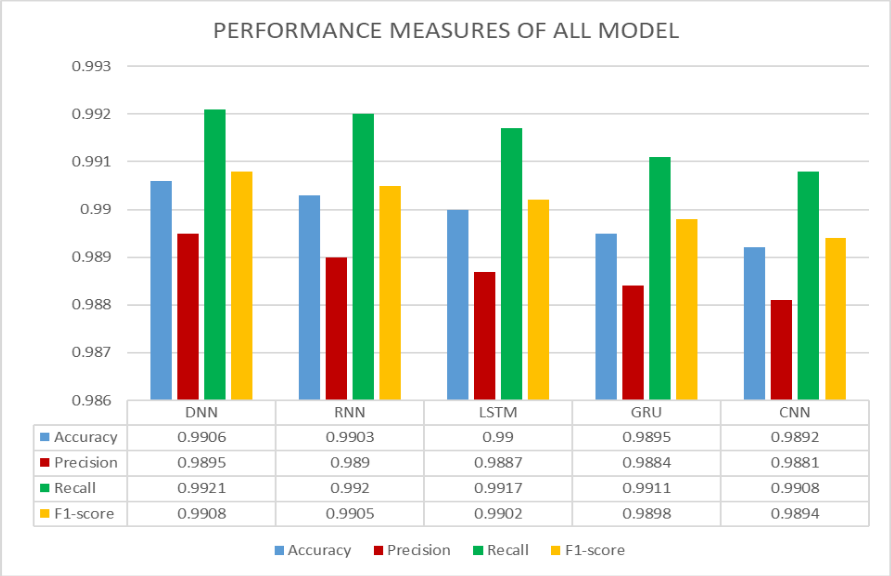


**Figure 3.** Our proposed models results.

### 4.3. Comparison Of The Proposed Model

The results I provided indicate that I have implemented and evaluated five different deep learning models for intrusion detection systems (IDS), including a DNN, RNN, LSTM, GRU, and CNN model. The best results for accuracy, precision, recall, and F1-score were obtained for the DNN model, followed by the RNN model, then the LSTM model, then the GRU model, and finally the CNN model as shown in figure 4.



| | DNN | RNN | LSTM | GRU | CNN |
|---|---|---|---|---|---|
| 🟦 Accuracy | 0.9906 | 0.9903 | 0.99 | 0.9895 | 0.9892 |
| 🟥 Precision | 0.9895 | 0.989 | 0.9887 | 0.9884 | 0.9881 |
| 🟩 Recall | 0.9921 | 0.992 | 0.9917 | 0.9911 | 0.9908 |
| 🟨 F1-score | 0.9908 | 0.9905 | 0.9902 | 0.9898 | 0.9894 |

**Figure 4.** Comparisons between our models.

### 4.4. Discussion

In this section we compare our proposed method with previous methods by researchers that have been published in articles. According to the results in table 1, it turns out that our method has proven high efficiency in training, testing and accuracy in prediction compared to other methods by researchers.

**Table1.** Comparison to previous works.

| Reference | Approach | Accuracy |
|---|---|---|
| Our Approach | DNN,<br>RNN,<br>LSTM,<br>GRU,<br>CNN | 99.6%<br>99.3%<br>99%<br>98.95%<br>98.92% |
| (Hodo, 2016) | FCNN | 99.4% |
| (Pajouh, 2016) | Naïve Bayes<br>and | 84.86 |

| | | |
|---|---|---|
| | CF-KNN | |
| (Tang, 2019) | RNN-GRU | 83.5% |

## 5. Conclusion

In conclusion, we have implemented a NIDS For SDN with IoTs based on DL by using various types of models including DNN, CNN, GRU, LSTM, and RNN. The system takes in input from an NSL-KDD dataset and enables the user to select the type of run, dataset, and model to use. The proposed system uses KERAS, TensorFlow and other library to implement the models and Sklearn library to examine the performance of the models. The performance of the models is measured by recall, precision, accuracy, F1-score, and confusion matrix. The best model is the one that has the highest recall, precision, accuracy and F1-score and it was DNN model. However, the model's performance will depend on the quality and quantity of data, and the specific problem we are trying to solve. Overall, we have implemented a DL-based NIDS and SDN with IoTs that can be used as a tool for detecting anomalies and intrusions in the network. In addition to what I previously mentioned, it's important to note that deep learning-based network intrusion detection systems have many advantages over traditional methods such as rule-based systems. They can handle high-dimensional and complex data, and can detect unknown intrusions that traditional methods may miss. Additionally, as the network security threat landscape is constantly evolving, deep learning-based systems can adapt to changing situations and improve their performance over time. However, it's also important to keep in mind that deep learning-based systems have some limitations as well. They require large amounts of labeled data to train the models, and may not be able to detect stealthy or targeted attacks. Additionally, the performance of deep learning-based systems can be affected by the quality of the data and the choice of architecture and hyperparameters. Therefore, it's important to carefully evaluate the performance of deep learning-based systems using appropriate evaluation metrics, and to continuously monitor and update the system to ensure that it remains effective in detecting intrusions. Additionally, it's important to use multiple layers of defense, such as intrusion detection systems, firewalls, and security information and event management (SIEM) systems, to provide comprehensive protection against network intrusions.

**References**

Ambika, N. (2021). A Reliable IDS System Using Blockchain for SDN-Enabled IIoT Systems. In IoT Protocols and Applications for Improving Industry, Environment, and Society (pp. 173-194). IGI Global.

Saritha, A., Ramasubba Reddy, B., & Suresh Babu, A. (2022). A Hybrid SDN Architecture for IDS Using Bio-Inspired Optimization Techniques. Journal of Interconnection Networks, 22(Supp01), 2141028.

Hendrawan, H., Sukarno, P., & Nugroho, M. A. (2019, July). Quality of service (qos) comparison analysis of snort ids and bro ids application in software define network (sdn) architecture. In 2019 7th International Conference on Information and Communication Technology (ICoICT) (pp. 1-7). IEEE.

Li, H., Wei, F., & Hu, H. (2019, March). Enabling dynamic network access control with anomaly-based IDS and SDN. In Proceedings of the ACM international workshop on security in software defined networks & network function virtualization (pp. 13-16).

Varghese, J. E., & Muniyal, B. (2021). An Efficient IDS framework for DDoS attacks in SDN environment. IEEE Access, 9, 69680-69699.

Ong, L. Y. (2014). OpenFlow/SDN and optical networks. Network Innovation Through OpenFlow and SDN: Principles and Design.

Raj, P., Raman, A., Raj, P., & Raman, A. (2018). Software-defined network (SDN) for network virtualization. Software-Defined Cloud Centers: Operational and Management Technologies and Tools, 65-89.

Sutton, R., Ludwiniak, R., Pitropakis, N., Chrysoulas, C., & Dagiuklas, T. (2021, April). Towards an SDN assisted IDS. In 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-5). IEEE.

Zwane, S., Tarwireyi, P., & Adigun, M. (2019, November). A Flow-based IDS for SDN-enabled Tactical Networks. In 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC) (pp. 1-6). IEEE.

Usman, S., Winarno, I., & Sudarsono, A. (2020, September). Implementation of SDN-based IDS to protect Virtualization Server against HTTP DoS attacks. In 2020 International Electronics Symposium (IES) (pp. 195-198). IEEE.

Alhowaide, A., Alsmadi, I., & Tang, J. (2021). Ensemble detection model for IoT IDS. Internet of Things, 16, 100435.

Panigrahi, A., Sahu, B., & Mohanty, S. N. (2022). A Survey on Opportunity and Challenges of IDS Over IoT. In Real-Life Applications of the Internet of Things: Challenges, Applications, and Advances (pp. 55-83). CRC Press.

Romeo, M. D., Rahman, N. A. A., & Yusof, Y. (2019). Intrusion Detection System (IDS) in Internet of Things (IoT) Devices for Smart Home. International Journal of Psychosocial Rehabilitation, 23(4).

Doshi, R., Apthorpe, N., & Feamster, N. (2018, May). Machine learning ddos detection for consumer internet of things devices. In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 29-35). IEEE.

Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, 82, 761-768.

Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P. L., Iorkyase, E., Tachtatzis, C., & Atkinson, R. (2016, May). Threat analysis of IoT networks using artificial neural network intrusion detection system. In 2016 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-6). IEEE.

Pajouh, H. H., Javidan, R., Khayami, R., Dehghantanha, A., & Choo, K. K. R. (2016). A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks. IEEE Transactions on Emerging Topics in Computing, 7(2), 314-323.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3), 12-22.

Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications surveys & tutorials, 18(2), 1153-1176.

Khodjaeva, M., Obaidat, M., & Salane, D. (2019). Mitigating Threats and Vulnerabilities of RFID in IoT Through Outsourcing Computations for Public Key Cryptography. Security, Privacy and Trust in the IoT Environment, 39-60.

Wrona, K., Amanowicz, M., Szwaczyk, S., & Gierłowski, K. (2017, May). SDN testbed for validation of cross-layer data-centric security policies. In 2017 International Conference on Military Communications and Information Systems (ICMCIS) (pp. 1-6). IEEE.

Shin, I., Choi, Y., Kwon, T., Lee, H., & Song, J. (2019, August). Platform design and implementation for flexible data processing and building ML models of IDS alerts. In 2019 14th Asia Joint Conference on Information Security (AsiaJCIS) (pp. 64-71). IEEE.

Nam, K., & Kim, K. (2018, October). A study on sdn security enhancement using open source ids/ips suricata. In 2018 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 1124-1126). IEEE.

Parkar, P., & Bilimoria, A. (2021, May). A survey on cyber security IDS using ML methods. In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 352-360). IEEE.

Kiran, U. (2022, January). IDS to detect worst parent selection attack in RPL-based IoT network. In 2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS) (pp. 769-773). IEEE.

Tang, T. A., McLernon, D., Mhamdi, L., Zaidi, S. A. R., & Ghogho, M. (2019). Intrusion detection in sdn-based networks: Deep recurrent neural network approach. Deep Learning Applications for Cyber Security, 175-195.

Alshahrani, A., & Clark, J. A. (2022, October). Transfer Learning Approach to Discover IDS Configurations Using Deep Neural Networks. In 2022 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI) (pp. 1-8). IEEE.

Boné, R., & Cardot, H. (2011). Advanced Methods for Time Series Prediction Using Recurrent Neural Networks. Recurrent Neural Networks for Temporal Data Processing, 15-36.

Wrona, K., Amanowicz, M., Szwaczyk, S., & Gierłowski, K. (2017, May). SDN testbed for validation of cross-layer data-centric security policies. In 2017 International Conference on Military Communications and Information Systems (ICMCIS) (pp. 1-6). IEEE.

Salem, F. M., & Salem, F. M. (2022). Gated RNN: The Gated Recurrent Unit (GRU) RNN. Recurrent Neural Networks: From Simple to Gated Architectures, 85-100.

Dai, J. (2021). Predicting machine's performance data using the stacked long short-term memory (LSTM) neural networks.

Meena, G., & Choudhary, R. R. (2017, July). A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA. In 2017 International Conference on Computer, Communications and Electronics (Comptelix) (pp. 553-558). IEEE.

Rahim, R., Ahanger, A. S., Khan, S. M., & Ma, F. (2022). Analysis of IDS using feature selection approach on NSL-KDD dataset.

Sharma, S., Gigras, Y., Chhikara, R., & Dhull, A. (2019). Analysis of NSL KDD dataset using classification algorithms for intrusion detection system. Recent Patents on Engineering, 13(2), 142-147.

Fong, J. S., Wong Ting Yan, K., Fong, J. S., & Wong Ting Yan, K. (2021). Information Systems Reengineering, Integration, and Normalization (pp. 1-28). Springer International Publishing.

Chen, B., & Ji, P. (2012). Numericalization of the self adaptive spectral rotation method for coding region prediction. Journal of Theoretical Biology, 296, 95-102.

Shou, Z., & Li, S. (2018). Large dataset summarization with automatic parameter optimization and parallel processing for local outlier detection. Concurrency and Computation: Practice and Experience, 30(23), e4466.

Wuisan, J. A., Jacobus, A., & Sompie, S. (2022). Data Balancing Methods on Radiographic Image Classification on Unbalance Dataset. Jurnal Teknik Elektro dan Komputer, 11(1), 1-8.