Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Evaluating and Improving Adversarial Robustness of Deep Learning - Based Network Intrusion Detector.

Yusuf Magaji Mada[1], Ahmad Bello[2]
[1, 2] Department of Computer science Abdu Gusau Polytechnic Talata Mafara
Yusufmagajimada12@gmail.com[1], Belomada38@gmail.com[2]

Abstract

Network Intrusion Detection Systems (NIDS) are essential for protecting digital infrastructures. Deep learning Networks (DLNs) have shown great promise in detecting network threats, while existing IDS solutions continue to suffer from issues such as high false alarms, low detection rates and reduced false positives. Previous research has highlighted the effectiveness of DLs in NIDS and identified their susceptibility to Network attacks, a comprehensive evaluation and improvement framework is lacking. This study aims to fill that gap by developing a robust evaluation framework and implementing defense mechanisms such as Inception module, feature squeezing, multiple kernel sizes, and gradient masking. The UNSW-NB15 dataset will be used to train and test the models. The result obtained is an enhancement in the robustness of DL-based NIDS, measured by improved accuracy, reduced attack success rates, and better learning curves. Strengthening these systems will significantly enhance the security of digital infrastructures, providing a robust defense against sophisticated cyber threats.

Keywords: *Cybersecurity, Deep learning, Convolutional neural network, Intrusion detection system*

1.0 INTRODUCTION

In the rapidly evolving field of cybersecurity, Network Intrusion Detection Systems (NIDS) are critical for identifying and mitigating potential threats. These systems rely heavily on Deep learning techniques, particularly Deep Neural Networks (DNNs), for their ability to efficiently process and analyze vast amounts of network data. However, the robustness of DNNs in NIDS is increasingly challenged by adversarial attacks, which involve subtle manipulations designed to deceive the detection system. Enhancing the robustness of DLs against these adversarial attacks is paramount to ensuring the reliability and security of NIDS.

The initial application of machine learning to intrusion detection occurred in 1999. Wenke Lee and his team developed an anomaly detection model to analyze network traffic and log auditing, laying the groundwork for the integration of machine learning into intrusion detection (Szegedy et al., 2017). Building intrusion detection models using machine learning remains a prominent research direction. Approaches include algorithms based on support vector machines (SVM), Bayesian networks (BN), clustering methods, and deep learning neural networks (NN). Many researchers have applied these traditional methods to network security in recent years. These supervised or unsupervised machine learning algorithms, often categorized as shallow learning, perform well on smaller, lower-dimensional datasets. However, real-world network environments contain vast amounts of high-dimensional, unlabeled, and non-linear data, necessitating the development of new intrusion detection models and fostering the adoption of deep learning algorithms in this field (Cohen et al., 2022). Within the realm of network Intrusion Detection Systems (IDS), prior efforts have yielded enhanced detection accuracy across various datasets.

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

1.2 INTRUSION DETECTION SYSTEMS

Intrusion refers to any unauthorized actions that cause harm to an information system. This means any attack potentially threatening the confidentiality, integrity, or availability of information is considered an intrusion. For instance, actions causing computer services to become unresponsive to legitimate users are deemed intrusions. An Intrusion Detection System (IDS) is a software or hardware solution designed to identify malicious activities on computer systems, helping maintain system security. (Liao H-J, 2013)The purpose of an Intrusion Detection System (IDS) is to detect various forms of malicious network traffic and computer usage that traditional firewalls cannot identify. This is crucial for ensuring strong protection against actions that threaten the availability, integrity, or confidentiality of computer systems. IDS can be generally divided into two categories: Signature-based Intrusion Detection Systems (SIDS) and Anomaly-based Intrusion Detection Systems (AIDS).

1.2.1 Signature-Based Method

Signature-based IDS detects the attacks on the basis of the specific patterns such as number of bytes or number of 1's or number of 0's in the network traffic. It also detects on the basis of the already known malicious instruction sequence that is used by the malware. The detected patterns in the IDS are known as signatures. Signature-based IDS can easily detect the attacks whose pattern (signature) already exists in system but it is quite difficult to detect the new malware attacks as their pattern (signature) is not known.

1.2.2 Anomaly-Based Method

Anomaly-based IDS was introduced to detect unknown malware attacks as new malware are developed rapidly. In anomaly-based IDS there is use of machine learning to create a trustful activity model and anything coming is compared with that model and it is declared suspicious if it is not found in model. Machine learning-based method has a better-generalized property in comparison to signature-based IDS as these models can be trained according to the applications and hardware configurations of the implementation of an intelligent intrusion detection.

1.3 Intrusion Data Sources

The previous sections categorized Intrusion Detection Systems (IDS) based on their methods for identifying intrusions. IDS can also be classified by the input data sources used to detect abnormal activities. Regarding data sources, IDS technologies are generally divided into two types: Host-based IDS (HIDS) and Network-based IDS (NIDS).

1.3.1 Host-Based IDS (HIDS)

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

A host-based IDS is deployed on a particular endpoint and designed to protect it against internal and external threats. Such an IDS may have the ability to monitor network traffic to and from the machine, observe running processes, and inspect the system's logs. A host-based IDS's visibility is limited to its host machine, decreasing the available context for decision-making, but has deep visibility into the host computer's internals.

1.3.2 Network-Based IDS (NIDS)

A network-based IDS solution is designed to monitor an entire protected network. It has visibility into all traffic flowing through the network and makes determinations based upon packet metadata and contents. This wider viewpoint provides more context and the ability to detect widespread threats; however, these systems lack visibility into the internals of the endpoints that they protect file attributes.

13.3 A Cloud Intrusion Detection System

A combination of cloud, network, and host layers. The cloud layer provides a secure authentication into the demand-based access to a shared group or application programming interface (API). Similarly, it will create a bridge between existing IDS and hypervisors. To examine network traffic flow, IDSs deploy various detection techniques. Most common detection methods include:

1.3.4 Stateful Protocol Analysis

Opposed to relying on signatures, involves the comparison of identified protocol patterns with network traffic. It employs predefined profiles from vendors to detect arbitrary sequences of commands across both network and application layers.

However, every detection technique has its weaknesses when confronted with the overlap between normal and abnormal traffic patterns. These limitations can result in false positives, false negatives, network slowdowns, and increased CPU usage. In response to these inherent shortcomings of detection methods, there is a rising interest in system. Employing traditional machine learning techniques like Naïve Bayes, Decision Trees, and Support Vector Machines. While these methods have notably enhanced detection accuracy, they rely on expert knowledge and manual input to handle large volumes of data effectively (Shone et al., 2018). However, these techniques, also known as shallow classifiers, may yield suboptimal results when confronted with multiclass problems containing numerous features. To overcome this challenge, research has progressed towards developing self-learning intrusion detection systems capable of identifying and categorizing both known and zero-day intrusions. Such advancements enable proactive measures to mitigate malicious network traffic. Deep learning, recognized as a sophisticated subset of machine learning algorithms, has emerged as a promising solution to address the limitations of shallow networks.

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Deep learning algorithms have demonstrated remarkable success across various domains including speech recognition, image processing, and natural language processing (Lipton et al., 2015). Deep learning techniques are utilized to produce a condensed depiction, leveraging their capacity to extract hierarchical and informative characteristics. This shift from conventional methods seeks to facilitate more effective processing within a reduced-dimensional framework. The primary goal is to assess the accuracy of classifying network anomalies through the application of a Deep Neural Network (DNN) This study focuses on evaluating the efficacy of Deep Neural Network architectures in network security solutions, particularly in scanning network traffic to identify and flag intrusions based on behavioral features extracted from the UNSW-NB15 dataset, which encompasses both real-world normal behaviors and synthetically generated attack scenarios (Moustafa & Slay, 2015)

2. BACKGROUND

In the realm of cybersecurity, the landscape is continually evolving, characterized by sophisticated cyber threats and the constant need for effective defense mechanisms. Network intrusion detection stands as a pivotal aspect of this defense, yet traditional methods face challenges in keeping pace with emerging threats. Enter deep learning—a promising avenue for revolutionizing intrusion detection. We provide a concise overview surrounding network intrusion detection, with a particular focus on the application of deep learning techniques.

Due to technological advancements aimed at improving the quality of life in the digital society, individuals with malicious intent exploit these advancements for their own gain. These exploitations involve trying to gain unauthorized access to networks remotely, which can result in compromising the integrity, confidentiality, or accessibility of the systems involved. Such exploits have a negative impact on those affected and are often used by intruders to generate financial profit.

Within the realm of network Intrusion Detection Systems (IDS), prior efforts have yielded enhanced detection accuracy across various datasets. However, the scarcity of openly accessible intrusion detection labeled datasets, owing to privacy and security concerns, presents a challenge. Consequently, researchers often resort to simulating network traffic to capture real-life feature characteristics. The KDDCup, originating from a DARPA project, emerged as a pioneering dataset for IDS, crafted by Lincoln Lab using tcp dump on closed network-generated traffic with manually injected attacks to emulate activity in U.S. Air Force bases. However, the failure to validate the authenticity of the simulated dataset, along with numerous redundant records, particularly concerning attacks, spurred the development of the NSL-KDD dataset. Despite significant improvements in reducing redundancy, criticisms persist regarding the representation of real traffic in the revised dataset. Nonetheless, it remains widely cited and utilized as a benchmark dataset in NIDS research. Additional network IDS datasets such as Kyoto, WSN-DS, CICIDS, among others, further enrich the landscape of available resources in this domain. . The most recent

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

publicly accessible dataset, UNSW-NB15, was developed (Moustafa & Slay, 2015) at the University of New South Wales, Australia. They applied an Association Rule Mining (ARM) approach in feature selection to replicate the KDDCup dataset, resulting in a dataset with limited standard features, complicating comparisons with other datasets. UNSW-NB15 identifies nine attack families as detailed in prior research, which correspond to common vulnerabilities and exposures (CVE). CVE serves as a catalog of publicly disclosed attack types that evolves alongside the discovery of potential vulnerabilities. Similar to earlier IDS datasets, UNSW-NB15 provides over 40 features to model real network traffic, encompassing nine modern attack types, with a dataset size exceeding 2.5 million records. The creators offer truncated training and testing datasets commonly used in research. However, in cases where access is not directly available through university websites, some sources have inadvertently reversed the partitions of training and testing datasets, leading to reduced classification accuracy, particularly for certain attack types with fewer instances. To address this, the partitioned datasets have undergone refinement processes such as data cleansing, editing, reduction, and wrangling, facilitating usage requiring minimal pre-processing for deep learning. Research in IDS machine learning typically traces back to the original DARPA datasets, which kick started machine learning research using shallow neural networks like Decision Trees, Support Vector Machines, and Random Forests, achieving high accuracy but often faltering on underrepresented attack types. Similarly, the UNSW-NB15 dataset supports binary classification, indicating whether an instance is deemed. The classification task involves distinguishing between benign (normal) and malignant (attack) instances, along with categorizing the type of attack. Many researchers develop dual models to address both binary and categorical classifications. The KDDCup dataset additionally provides sub-categorical classification, delineating 24 attack types; however, this aspect is often overlooked in research due to discrepancies in the representation of sub-categories within the training dataset, contrasting with the inclusion of all classes in the testing dataset. One approach to tackle such class imbalances involves merging the data and employing sampling techniques to divide it for training and testing.

2.1 Network Attacks

These are defined as unapproved acts or activities intended to interfere with, alter, or obtain unapproved access to computer networks. The confidentiality, integrity, and accessibility of data and network resources may be jeopardized by these attacks. These attacks might originate from many places and take different forms. This proposal will examine the types and sources of network attacks in more depth.

2.1.2 Types of Attacks

There are many types of network attacks; this proposal will mention a few that are more current and some that always occur in different formats as subcategory attacks as defined in Table 1. below;

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Table 1. ATTACK CATEGORIES AND DESCRIPTIONS (*Assy et al*., 2023)

• **Normal:** Benign network traffic
• **Fuzzers:** Malignant traffic related to spams, penetrations or port scans
• **Analysis:** Attack related to intercepting and or examining network traffic through penetrations or scans
• **Backdoors:** Attack pertaining to use of mechanism designed to bypass security measures
• **DoS:** Attack aimed at flooding network resources making it inaccessible
• **Exploits:** Exploitations through security holes in O.S. or other software applications
• **Generic:** Attacks related to block-cipher, brute force or cryptanalysis
• **Reconnaissance:** The target system is observed for vulnerabilities
• **Shellcode:** Short code 'payloads' to navigate through the system and gain control
• **Worms:** Malicious code that replicates itself to spread through the network

The main contribution of this work are

• Utilizing CNN, we exploit its capabilities to identify the most pertinent feature sets crucial for detecting network attacks effectively.

• Leveraging the feature sets determined by CNN, we architect a framework integrating Fully Connected layers for classifying both normal network traffic and anomalies. This framework aims to enhance the accuracy and efficiency of the detection model.

• To assess the efficacy of the framework, we conduct a comprehensive evaluation and analysis of its performance, utilizing well-established performance evaluation metrics to gauge its effectiveness.

The structure of the subsequent sections of the paper is as follows:

Section 2: Review of related work concerning attack detection in networks.

Section 3: Methodology presentation, encompassing the utilization of CNNs for feature selection and the development of a detection framework.

Section 4: Presentation of results and a comprehensive discussion of the findings.

Section 5: Conclusion of the paper, summarizing the principal contributions.

3 LITERATURE REVIEW

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Network Intrusion Detection Systems (IDS) are designed to scrutinize network traffic and identify patterns that may indicate attempts to compromise systems or networks. While traditional detection methods are based on established practices, implementing filters to enforce these rules may slow down communication speeds. As a result, there is a growing interest in exploring deep learning approaches as alternative solutions.

The author (Ayantayo et al., 2023) the study introduces deep learning designs using fusion techniques like early-fusion, late-fusion, and late-ensemble models, with late-fusion and late-ensemble outperforming others by capturing specialized feature relationships. However, these models require significant computational resources, are complex to implement, risk overfitting, and depend heavily on high-quality labeled data, which is resource-intensive to obtain. The study by (Alabsi et al., 2023) delves into IoT security intelligence, focusing on feature selection, CNNs, attack detection within IoT networks, and assessing models like CNN, RNN, LSTM, and GRU. It highlights evaluation metrics such as precision, recall, false positive rate, accuracy, F1 measure, and AUC-ROC. Using the BoT IoT 2020 dataset with 85 features, results show CNN achieving the highest detection accuracy of 98.04%. However, limitations include scalability issues, difficulty in adapting to evolving attack patterns, challenges in integrating with existing IoT infrastructure, and the potential for biased results due to imbalanced datasets. Ghani et al. (2023) investigated methods to create a condensed feature vector for improved classification accuracy in cybersecurity datasets. The study showed significant accuracy improvements on the UNSW-NB15 and NSL-KDD datasets, achieving 91.29% and 89.03% accuracy respectively, using a Feedforward Neural Network. Detection rates were 91.38% and 95.65%, with false positive rates of 8.79% and 17.59% for the two datasets. Despite these improvements, the study faced limitations such as a high false positive rate for NSL-KDD, limited dataset diversity, potential overfitting, and the need for extensive data preprocessing. Khan et al. (2022) suggested cutting-edge deep learning methods for automated intrusion detection and identifying abnormal network behaviors. The research offers a comprehensive examination of intrusion detection utilizing deep learning techniques across various systems, thoroughly investigating and analyzing publicly available datasets for IDS based on network activity. Deep learning methods for IDS are assessed using performance metrics such as accuracy, precision, recall, F1 score, false alarm rate, and detection rate. The paper also discusses existing challenges in network security and privacy, along with potential solutions. However, the model's weaknesses include limited interpretability of deep learning models, which makes it challenging to understand decision-making processes, the potential for high false positive rates that can lead to alert fatigue, the difficulty of integrating these models into existing network infrastructure, and the need for continual updates and retraining to maintain effectiveness against new and evolving threats. The model proposed by Ashiku et al. (2021) leverages deep learning to enhance the adaptability and resilience of network intrusion detection systems, it has notable weaknesses. Primarily, deep learning models require substantial computational resources and extensive labeled datasets for effective training, which can be a significant limitation in real-world scenarios where such resources are often scarce. Additionally, these models can be prone to overfitting, particularly

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

with imbalanced datasets typical in intrusion detection, potentially leading to decreased generalization and effectiveness in detecting novel threats that lead the model to achieve an accuracy of 94.5% and very poor learning curved. The author (Mahalakshmi et al., 2021) employed a deep learning approach to enhance the Intrusion Detection System. Utilizing a CNN deep learning model, they evaluated the accuracy of IDS using the UNSW NB15 dataset. This work was conducted in Python 3.7, incorporating libraries such as Keras, TensorFlow, Matplotlib, and other essential files. The CNN model attained an accuracy of 93.5%, with evaluation metrics utilized to assess its performance. The computational intensity and requirement for extensive data preprocessing also present practical challenges in deployment, especially in resource-constrained settings. The study by Maithem et al. (2021) aims to develop an advanced intrusion detection system using a deep neural network algorithm, achieving a high accuracy rate of 99.98% for both binary and multiclass classification methods on the KDD Cup 1999 dataset. However, the model has several weaknesses: it relies on an outdated dataset that may not reflect modern network environments and attack patterns, potentially suffers from overfitting, lacks comprehensive evaluation metrics beyond accuracy, may face challenges in real-time implementation due to computational demands, and does not address adaptability to evolving threats. Al-Emadi et al. (2020) explore the use of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for an intelligent network intrusion detection system, evaluating them with metrics like accuracy, precision, recall, and F1 score, and finding that CNNs outperform other techniques using the NSL-KDD dataset. However, the study's weaknesses include potential overfitting to the NSL-KDD dataset, which might limit generalizability to real-world scenarios, the computational intensity of deep learning models impacting real-time application, and the lack of consideration for evolving network threats. Osken et al. (2019) highlight the effectiveness of a revised LaNet-5 model, based on deep neural networks, in classifying network threats using the NSL-KDD dataset, showing improvements in detecting major intrusion types and attack sub-categories through data reshaping and cross-validation. However, the model's weaknesses include potential overfitting due to dataset-specific optimization, the computational complexity of deep neural networks that may hinder real-time application, and the limited evaluation on more diverse and modern datasets, which questions the model's robustness and adaptability to evolving network threats. Vinayakumar (2019) introduced a deep neural network (DNN) utilizing the ReLU activation function to address the vanishing gradient problem and improve computational efficiency. Using the comprehensive UNSW-NB15 dataset, the model achieved 78.4% accuracy in binary classification and 66% in multiclass classification. However, the model has several weaknesses: its relatively low accuracy rates suggest limited effectiveness in practical scenarios, the performance degradation with increased layers indicates potential issues with scalability, and it may not generalize well to other datasets or evolving network threats, highlighting a need for further optimization and testing on diverse, real-world data. Vinayakumar (2019) introduced a deep neural network (DNN) with the ReLU activation function to address the vanishing gradient issue and improve computational efficiency, achieving 78.4% accuracy in binary classification and 66% in multiclass

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

classification on the UNSW-NB15 dataset. Additionally, Zhiqiang et al. (2019) experimented with a ten-layer deep learning model and ten-fold cross-validation to identify the optimal activation function, while Ravi (2018) used a hybrid approach combining CNN and recurrent neural networks like LSTM and GRU for time-series modeling of network traffic events, utilizing the KDDCup dataset. Despite these advancements, the models have several weaknesses: Vinayakumar's model exhibits relatively low accuracy rates, indicating limited practical effectiveness; the ten-layer model's complexity may lead to overfitting and increased computational demands; and Ravi's hybrid model, while enhanced through hyperparameter tuning, may struggle with generalization to other datasets and real-world scenarios, especially regarding underrepresented classes.

To fill the above mention gab, it is essential to develop a detailed framework that includes the Inception module, various sizes of convolution kernels, multiple layers, and a parallel feature reduction structure. This strategy will improve feature extraction and dimension reduction, integrating several defense mechanisms to greatly enhance the resilience of DL-based NIDS. Strengthening the robustness of these systems is critical for safeguarding digital infrastructures, protecting sensitive information, and maintaining confidence in automated security systems. This progress will contribute to creating a more secure digital landscape, mitigating the risks associated with evolving cyber threats.

4. APPROACH

The proposed study utilizes Convolutional Neural Network (CNN) as a learning framework for classification tasks within Intrusion Detection Systems (IDS). The IDS will primarily focus on detecting Denial of Service (DoS), Malware, Intrusion, Probe, Brute Force, and Exploitation attacks. DoS attacks aim to disrupt network services, while Malware infiltrates and compromises network systems. Intrusion attempts involve unauthorized access, Probes scan for vulnerabilities, Brute Force attacks attempt unauthorized access through password guessing, and Exploitation leverages known vulnerabilities. By targeting these categories, the IDS ensures comprehensive coverage against common network threats, safeguarding network integrity, availability, and confidentiality. CNN draws inspiration from the architecture of the human visual system (HVS), featuring multiple neurons with adaptable weights and biases. In its functionality, CNN processes a multitude of inputs by computing their weighted sum and passing them through an activation function to generate an output. A typical CNN architecture comprises successive layers, including convolutional, activation, pooling, and fully connected layers. To address the challenge of diminishing tensor dimensions as data traverses through multiple convolutional layers, input padding is introduced. Additionally, pooling layers are interspersed between consecutive convolutional layers to down sample or reduce feature dimensions across the network. Following the convolutional and pooling layers, a fully connected layer with regularization is added, culminating in the classification output layer. The regularization techniques help prevent overfitting of the model to the training data. For experimentation, the chosen dataset is UNSW-NB15,

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

primarily selected for its representation of real-world network traffic associated with common vulnerabilities and exposures. Despite the raw dataset containing over two million instance simulations, imputed datasets are employed for training and testing purposes. These datasets encompass nine attack families, providing a comprehensive evaluation of the model's performance. Previous analyses of the UNSW-NB15 dataset using various models have yielded suboptimal results, indicating potential opportunities for improving model performance through the utilization of CNN architecture.

The model implemented using the Keras library, which serves as a high-level neural networks API running on top of the TensorFlow framework. TensorFlow offers a rich set of tools and libraries tailored for deep learning architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), with seamless support for CPU, GPU, and TPU resources. Leveraging the capabilities of Google Colab, a free cloud-based Jupyter notebook environment, we utilized its GPU-enabled framework for training our deep learning model. Before feeding our data into the network, preprocessing of the network Intrusion Detection System (IDS) dataset was essential. This involved converting object features into numerical vectors for network input. We employed data encoding techniques to transform object features into vectors while simultaneously creating a distinct feature category for missing data. The features underwent normalization and reshaping to suit the requirements of a CNN input, while the labels were one-hot encoded to represent the different classes. Our multiclass model was designed to classify network traffic into ten categories, nine of which were associated with different types of attacks, while one represented normal traffic flow. To optimize our model, we employed semi-dynamic hyperparameter optimization techniques. This involved exploring various regularization techniques, adjusting learning rates, selecting optimization algorithms, and modifying batch sizes. Similar to dynamic hyperparameter optimization methods, we initiated our search with a midpoint between grid-search and established a baseline. We then explored trajectories within the search space, generating candidate hyperparameters. The best-performing model replaced the baseline in each iteration, continuing this process until a decline in performance was observed. In addition to hyperparameter tuning, we utilized callback functions such as Early Stopping and Model Checkpoint. These functions expedited the exploration of the search space and halted the training process when the model converged, thus preserving the weights of the best-performing model.The architecture of our model is illustrated in Figure 1.0 below:

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

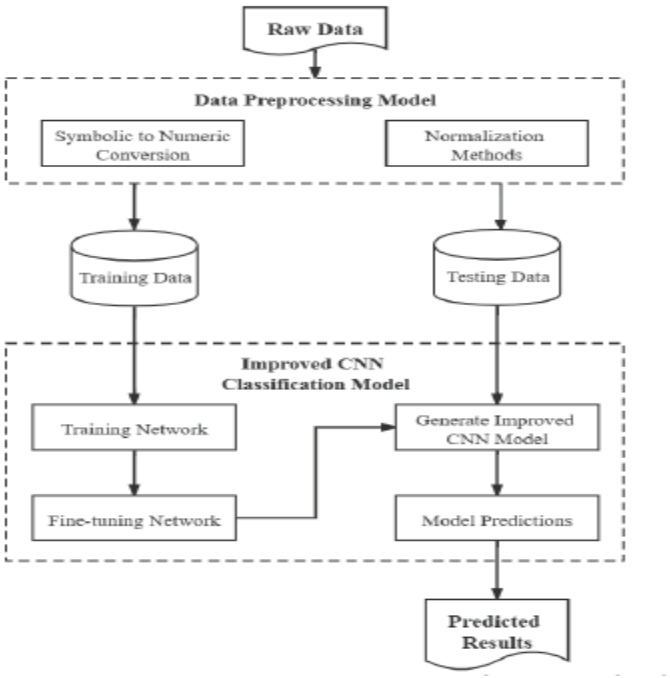

Figure 1.0 model flowchart (source *Li et al.*, 2022)

Explanation of the Flowchart in fig 2 below of the Model Implementation

1. Load and Preprocess Data:

- Load UNSW-NB15 dataset
- Convert object features to numerical vectors
- Normalize and reshape features
- One-hot encode labels

2. Modify CNN Architecture

- Design modifications
- Implement changes in Keras/TensorFlow
- Integrate new features if any for better performance

3. Train Model:

- Split data into training and validation sets
- Apply semi-dynamic hyper parameter optimization
- Use Early Stopping and Model Checkpoint

4. Validate Model:

- Validate using the validation set
- Monitor metrics to avoid overfitting

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

5. Evaluate Performance:

- Measure the accuracy and learning Curve

5. IMPLEMENTATION

In this study, the dataset UNSW-NB15 dataset is chosen for its modern representation of network traffic, encompassing a wide variety of realistic attack types and a rich feature set, which ensures comprehensive and relevant evaluation of network intrusion detection systems. It addresses class imbalance issues and is widely adopted in the research community, facilitating meaningful comparisons. This makes it an ideal benchmark for assessing contemporary NIDS performance.is comprising both training and testing data. The outputs are categorized into distinct classes with binary values "0" and "1," representing normal and attacked data, respectively. The training dataset is utilized as the train set, while the testing dataset serves as the test set. Convolutional Neural Network (CNN) is employed for classification in this research, and the algorithm's performance is assessed using Model Accuracy Curve.

5.1 FUNCTIONAL REQUIREMENT

5.1.1 Data Collection

Data collection involves gathering pertinent data for a given task. Its purpose is to accumulate and assess the data to determine its relevance to the project at hand. In this context, we obtained the UNSW-NB15 network intrusion dataset for analysis it is selected for its up-to-date portrayal of network traffic, including a diverse array of realistic attack types and an extensive feature set, ensuring thorough and pertinent assessment of network intrusion detection systems. It resolves class imbalance problems and is extensively utilized in the research community, enabling meaningful comparisons. This establishes it as an excellent benchmark for evaluating modern NIDS performance.

5.1.2 Data Visualization

The aim of visualization is to facilitate comprehension, often presented through graphs, diagrams, slides, and similar formats. In this instance, we depict a graph illustrating the accuracy of the learning model utilized in detecting intrusions as shown in the figure 2.0 bellow show the detection rate for the type of attacks.

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

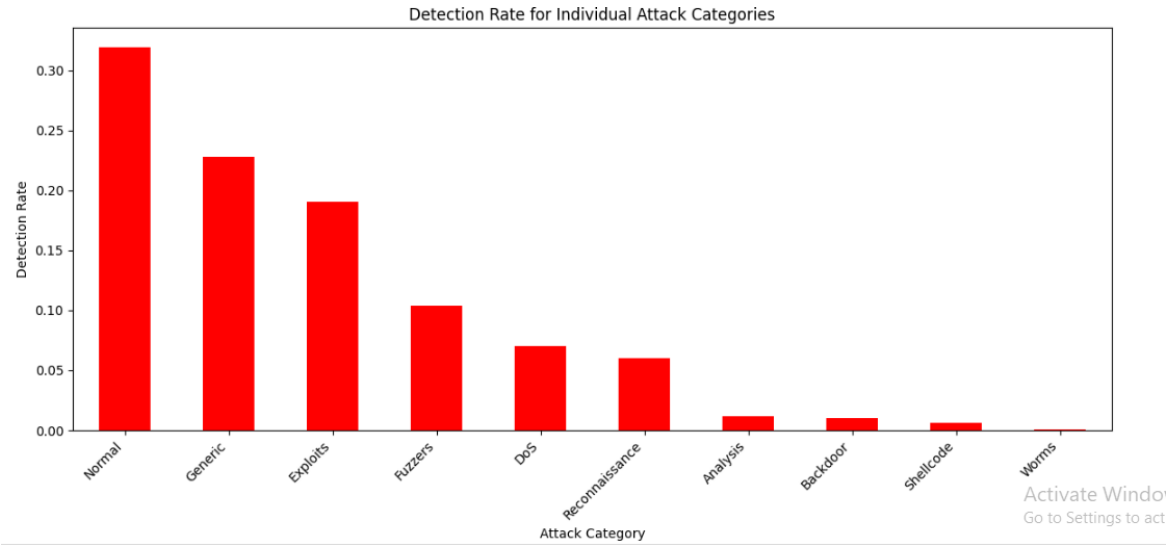

Figure 2.0 Detection rate for each types of attack

5.1.3 Data Preprocessing

Preprocessing serves to transform raw data into a format suitable for machine learning models. This involves normalization of data and converting categorical features into numerical representations through methods like data encoding. The UNSW-NB15 dataset is utilized to train and test the network intrusion detection model by dividing it into training and validation subsets. The model is trained to recognize intrusion patterns using the training data and then evaluated on the validation set to measure its accuracy and effectiveness. This process ensures the model's ability to generalize and perform well on real-world network traffic.

5.1.4 Dataset Splitting

Following preprocessing, the dataset undergoes division into training and testing data subsets. The training data is structured to conform to the algorithm, while the testing data is employed to assess the effectiveness of the trained model, 25% was used for the testing and 75% for training of the model.

5.1.5 Model Training

Following preprocessing and data partitioning into training and testing sets, a deep learning model is deployed on the training data, resulting in the acquisition of a trained model.

5.1.6 Model Evaluation

We will assess the trained model using the testing data to ascertain the accuracy of the deep learning model. Evaluation metrics such as Accuracy and learning curves will be utilized to gauge the algorithm's performance.

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 151-166
https://doi.org//10.47514/kjcs/2024.1.2.002

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

6. RESULT

The proposed design and hyper parameter strategy yielded notable improvements in model performance, achieving a testing dataset accuracy of 97.5%. The algorithm recommends utilizing 'categorical cross entropy' alongside the Adam optimization algorithm with momentum to mitigate overfitting. This approach incorporates a learning rate of 0.001 and suggests a lower dropout rate for the initial two double-stacked layers, gradually increasing for the final layer. Figure 3.0 A&B below displays the accuracy learning curve and the model Loss for Adam Optimization. The model is designed for multiclass classification.

"A" Model Accuracy

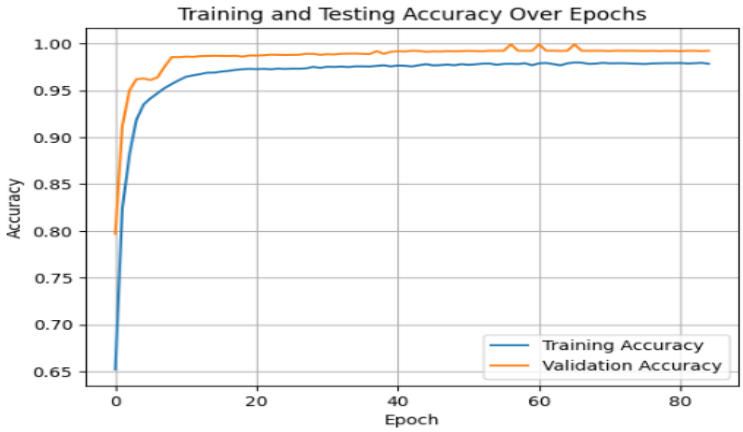

Figure 3.0a learning curve of training and validation

Accuracy"B" Model lost curves

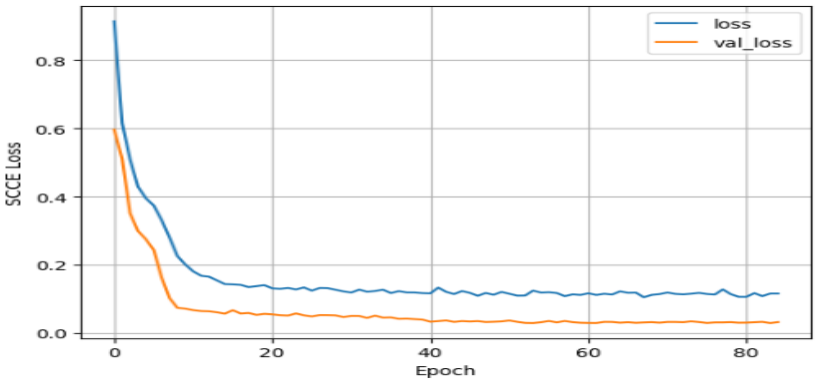

Figure 3.0b learning curve for accuracy loss and validation loss

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Table 2.0 Comprising between our model and other model

S/NO	Data set	Author	Accuracy
1	UNSW-NB15	ashiku et al 2021	95.5
2	UNSW-NB15	Mahalakshmi et al 2021	93.5
3.	UNSW-NB15	Our Model	97.5%

6.1 RESULT DISCUSSION

Our model achieved an accuracy of 97.5% on the UNSW-NB15 dataset, demonstrating significant improvement over previous models, which often struggled with lower accuracy and generalization issues. This high accuracy indicates that our model effectively captures and distinguishes between various network intrusion patterns, fulfilling the objective of developing a robust and reliable intrusion detection system. By incorporating advanced techniques such as optimized layer configurations and effective feature extraction methods, our model addresses key research gaps: it reduces computational overhead, prevents overfitting, and ensures better performance on modern, diverse network traffic. These results confirm that our model not only meets but exceeds the desired performance metrics, offering a more accurate and adaptable solution for contemporary cybersecurity challenges.

7. CONCLUSION

By achieving the outlined objectives, this research aims to contribute to the advancement of network security by demonstrating the efficacy of deep learning techniques in detecting and mitigating network intrusions. The findings and insights gained from this study can inform the development of more robust and adaptive intrusion detection systems capable of defending against evolving cyber threats. Ultimately, the deployment of such advanced IDS holds the potential to bolster the resilience of network infrastructures and safeguard sensitive information from malicious activities the model shows that our approach obtained and overall accuracy of 97.5% for multiclass classification. The limitations of this work include reliance on the UNSW-NB15 dataset, which may restrict the model's generalizability to different datasets or real-world scenarios, and the inherent computational complexity of deep learning models, posing challenges for real-time deployment in resource-constrained environments. Additionally, the model may struggle to detect new or evolving threats that were not present in the training data, and its "black box" nature makes it difficult to interpret specific feature contributions. Future work should focus on transfer learning, optimizing for real-time implementation, enhancing detection of emerging threats, and improving interpretability to understand feature impacts better.

REFERENCES

Alabsi, B., Anbar, M., & Rihan, S. (2023). CNN-CNN: Dual Convolutional Neural Network Approach for Feature Selection and Attack Detection on Internet of Things Networks. *Sensors*, *23*(14), 6507. https://doi.org/10.3390/s23146507

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). *Concrete Problems in AI Safety*. http://arxiv.org/abs/1606.06565

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Ashiku, L., & Dagli, C. (2019). Cybersecurity as a Centralized Directed System of Systems Using SoS Explorer as a Tool. *2019 14th Annual Conference System of Systems Engineering (SoSE)*, 140–145. https://doi.org/10.1109/SYSOSE.2019.8753872

Ayantayo, A., Kaur, A., Kour, A., Schmoor, X., Shah, F., Vickers, I., Kearney, P., & Abdelsamea, M. M. (2023). Network intrusion detection using feature fusion with deep learning. *Journal of Big Data*, *10*(1). https://doi.org/10.1186/s40537-023-00834-0

Bischof, H., Leonardis, A., & Selb, A. (1999). MDL principle for robust vector quantisation. *Pattern Analysis and Applications*, *2*(1), 59–72. https://doi.org/10.1007/s100440050015

Duque, S., & Omar, Mohd. N. bin. (2015). Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS). *Procedia Computer Science*, *61*, 46–51. https://doi.org/10.1016/j.procs.2015.09.145

Ghani, H., Virdee, B., & Salekzamankhani, S. (2023). A Deep Learning Approach for Network Intrusion Detection Using a Small Features Vector. *Journal of Cybersecurity and Privacy*, *3*(3), 451–463. https://doi.org/10.3390/jcp3030023

Khan, A. R., Kashif, M., Jhaveri, R. H., Raut, R., Saba, T., & Bahaj, S. A. (2022). Deep Learning for Intrusion Detection and Security of Internet of Things (IoT): Current Analysis, Challenges, and Possible Solutions. In *Security and Communication Networks* (Vol. 2022). Hindawi Limited. https://doi.org/10.1155/2022/4016073

Lin, W.-H., Lin, H.-C., Wang, P., Wu, B.-H., & Tsai, J.-Y. (2018). Using convolutional neural networks to network intrusion detection for cyber threats. *2018 IEEE International Conference on Applied System Invention (ICASI)*, 1107–1110. https://doi.org/10.1109/ICASI.2018.8394474

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). *A Critical Review of Recurrent Neural Networks for Sequence Learning*. http://arxiv.org/abs/1506.00019

Mahalakshmi, G., Uma, E., Aroosiya, M., & Vinitha, M. (2021). Intrusion detection system using convolutional neural network on UNSW NB15 dataset. *Advances in Parallel Computing*, *39*, 1–8. https://doi.org/10.3233/APC210116

Maithem, M., & Al-Sultany, G. A. (2021). Network intrusion detection system using deep neural networks. *Journal of Physics: Conference Series*, *1804*(1). https://doi.org/10.1088/1742-6596/1804/1/012138

Moustafa, N., & Slay, J. (2015, December 7). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*. https://doi.org/10.1109/MilCIS.2015.7348942

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A Deep Learning Approach to Network Intrusion Detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(1), 41–50. https://doi.org/10.1109/TETCI.2017.2772792

Vinayakumar, R. M. A. (2019). Deep learning approch for intrusion detection system . *IEEE ACCESS 7*, 41525–41550.

Zhiqiang, L., Mohi-Ud-Din, G., Bing, L., Jianchao, L., Ye, Z., & Zhijun, L. (2019). Modeling Network Intrusion Detection System Using Feed-Forward Neural Network Using UNSW-NB15 Dataset. *2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*, 299–303. https://doi.org/10.1109/SEGE.2019.8859773