Distributed Denial-of-Service (DDoS) Detection Using Multitask Learning based on Deep Learning

1st Muhammad Fauzan Abyandani
School of Computing
Telkom University
Bandung, Indonesia
fauzanbyan@student.telkomuniversity.ac.id

2nd Parman Sukarno
School of Computing
Telkom University
Bandung, Indonesia
psukarno@telkomuniversity.ac.id

3rd Aulia Arif Wardana
Faculty of ICT
Wrocław University of Science and Technology
Wrocław, Poland
aulia.wardana@pwr.edu.pl

***Abstract*—Distributed Denial of Service (DDOS) is one of the most significant threats among a wide variety of threats that can attack increasingly vulnerable computer networks. Traditional detection methods often fail to effectively manage the complexity of modern attack scales because traditional methods usually rely on monitoring the traffic volume and identifying spikes as an attack. This approach causes ineffective, inaccuracy, and lack of scalability. To address these challenges, this research aims to develop an advanced and innovative approach to detect and classify DDOS attacks. We use Multitask Learning (MTL) combined with Deep Learning (DL) using three DL models: Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN). Using the NF-CSE-CIC-IDS2018-V2 and NF-BoT-IoT-V2 datasets, our methodology includes data pre-processing, feature normalization, and adjustment to a specific model such as changing the input. The evaluation shows that MTL-CNN model achieves perfect scores across multiple metrics: 100% accuracy, 100% precision, 100% recall, and 100% F1-score, as well as an execution time of 32.77 seconds. Although there is an increase in the time metric because of MTL compared to Single Task Learning(STL), MTL combined with the DL algorithm allows the model to learn faster with shared representation. This allows faster learning over time using knowledge learned from both datasets, resulting in better generalization and stronger attack detection.**

***Index Terms*—Convolutional Neural Network(CNN), Deep Learning(DL), Distributed Denial of Service (DDOS), Intrusion Detection, Long Short-Term Memory(LSTM), Multi-Layer Perceptron(MLP), Multitask Learning(MTL)**

I. Introduction

One of the most significant threats to network and security is DDOS attacks, intended to overload the system and disrupt services by flooding traffic from various devices, making the service inaccessible. Traditional detection methods often lack the efficiency, accuracy, and scalability needed for complex and large-scale attacks. These challenges happened because traditional methods only focused on monitoring traffic volumes and detecting spikes as the indicator of an attack [1]. Current detection methods are inadequate, emphasizing the need for more robust and effective approaches to counter DDoS attacks, given the characteristics of this attack.

This research aims to develop an advanced approach to detect and classify DDoS attacks using MTL and DL. MTL can simultaneously learning if there are threats while classifying the type of attack, improving the efficiency and generalization [2]. Meanwhile, DL can recognize complex patterns used in DDoS attacks by automatically finding relevant information from the raw data [3]. DL can find complex patterns, making it an ideal choice for modern DDoS detection, compared to conventional machine learning, which depends on predefined features. This collaborative intrusion detection has the potential to improve the security of systems in terms of monitoring possible intrusions and abuse of Integrity. By using different datasets, appropriate training methods will yield optimal output and, hence, better detection capabilities [4] [5]. This study explores the use of MTL and DL in DDoS detection and classification as tools for evaluating the current techniques. The objective is to combine MTL and DL with their advantages to build a more efficient, scalable, and precise model.

This research fills a gap in previous studies by Albelwi, which highlights the need for more complex datasets and sophisticated algorithms that integrate MTL and DL [6]. To improve the detection and classification of DDoS attacks, this research used two datasets: NF-CSE-CIC-IDS2018-V2 and NF-BoT-IoT-V2. These datasets were chosen due to their implementation and advanced features that can handle the complexities of modern DDoS attacks. As a result, this research enhances DDoS feature detection and classification, overcoming the previous weaknesses to yield a more robust and coherent approach.

II. Related Works

Research [6] by Saleh Ali Albelwi. (2022) proposed an intrusion detection system (IDS) based on MTL and Deep Neural Networks (DNN) for simultaneous detection of multiple attack types. The model by Albelwi using UNSW-NB15 and CICIDS2017 datasets achieved 87.50% accuracy, which is better than traditional neural networks (81.48%) and decision tree (86.41%) models. Generalization can be achieved through this approach, as well as minimizing overfitting, which will make it possible to resist complex attacks on network security. This study, therefore, demonstrates how MTL could boost IDS performance. However, there is a need for further analysis of the model's capabilities because there are no measures in this

study to determine its precision, recall, F1 score, or execution time.

Research [7] by Shakya and Abbas. (2021) has conducted a comparative analysis among different machine learning models to detect DDoS attacks in Internet of Things (IoT) networks. The research investigates using performance metrics such as accuracy, precision, recall, and F1 score. In conclusion, it was found that the XGBoost model had better results with 99.82% accuracy, 99.8% precision, 99.85% recall, and finally, an F1 score of 99.82%, compared with a lower score for K-Nearest Neighbor (KNN) model. The XGBoost model performed better than KNN; hence, it may be suitable for complex and large datasets. However, no computational time was involved during these experiments, which are crucial in the real-time detection of DDoS scenarios.

Research [8] by Halladay et al. (2023) has conducted research involving the detection and classification of DDoS attacks using time-based features, with a performance comparison among various machine learning and DL algorithms. The Deep Neural Network (DNN) model achieved 99% accuracy, an F1-score of 100%, and a training time of 185.61 seconds. The Support Vector Machine (SVM) model also achieved 99% accuracy and an F1-score of 100% but took a longer training time of 203.28 seconds. This result shows that time-based characteristics can slightly reduce precision and training times, making them more useful for real-time applications. These outcomes represent an initial step towards efficient and effective DDoS detection.

Research [9] by Sarhan et al. (2022) proposed a Network Intrusion Detection System (NIDS) using a standardized NetFlow-based feature set in order to improve the robustness and consistency of ML-based evaluation across various datasets and attack scenarios. The NF-BoT-IoT-v2 dataset achieved 100% accuracy, 100% F1 score, and 3.90 µs prediction time, while the NF-CSE-CIC-IDS2018-v2 dataset achieved 99.35% accuracy, 97% F1 score, and 21.75 µs prediction time. The algorithm, along with metrics such as precision, recall, and execution time, were not provided in this study. This research highlights the usefulness of improving detection accuracy and prediction efficiency of ML-based NIDS by applying a NetFlow-based feature set.

The problem of detection and prevention of DDOS attacks has been widely studied. However, no known research has combined MTL and DL for DDOS detection using the datasets that are NF-CSE-CIC-IDS2018-V2 and NF-BoT-IoT-V2. On the other hand, some studies, such as S. Ali et al. (2023) and Q. Liu et al. (2022), have investigated MTL and DL for tasks similar to those in this study, but they cannot be directly compared. Ali's research focused on detecting malware using behavioral traffic analysis on IoT devices, considering different datasets, and identifying malware that did not detect DDoS [10]. However, Liu's research used MTL for intrusion detection. Moreover, they utilized different datasets and did not prioritize execution time or provide a thorough evaluation of several DL methods, including MLP, LSTM, and CNN [11]

The current study differs from others as it addresses the high data complexity of NF-CSE-CIC-IDS2018-V2 and NFBoT-IoT-V2 datasets, which still conduct execution time optimization and performance evaluation with respect to several DL models. The MTL approach allows the model to capture inter-task relationships simultaneously, leading to efficient data processing. This integration between DL and MTL facilitates automatic feature extraction from raw data, thereby enhancing sensitivity and accuracy in detecting and classifying numerous DODS attacks. This technique improves the model's performance and overcomes significant challenges arising from the data's complex nature and diverse attack patterns.

III. Methodology

The system design for this research is illustrated in Fig. 1 through a flowchart.

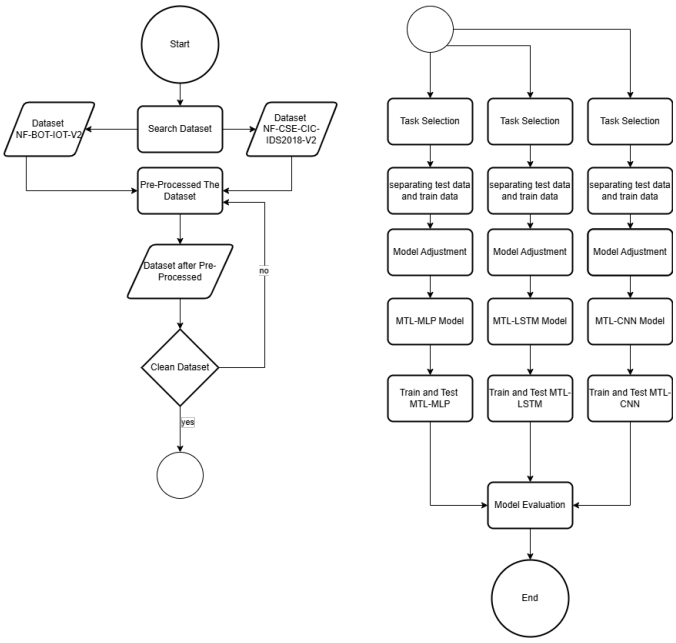

Fig. 1. Flowchart System

The process begins by preparing the datasets and then preprocessing the data to ensure it is clean and ready for model testing. From each dataset, 100,000 samples are selected and filtered for intrusion detection, specifically focusing on DDoS detection. Tasks are then assigned, and the data is split into 70% for training and 30% for testing. A DL-based MTL model is created, trained with the prepared training data, and assessed using the test data. The evaluation metrics for assessing the model's performance include accuracy, precision, recall, F1-Score, and execution time. These metrics are crucial for measuring the model's classification performance, the balance between precision and recall, and efficiency in terms of execution time.

A. Datasets Description and Preprocessing

This research utilizes two datasets: NF-BoT-IoT-V2 and NFCSE-CIC-IDS2018-V2. The NF-BOT-IOT-V2 dataset, developed by the University of Queensland, contains 30,420,085

entries focused on network traffic involving botnets on IoT devices. The NF-CSE-CIC-IDS2018-V2 dataset from CSE-CIC includes 17,129,714 entries aimed at detecting intrusion attacks such as DDoS and port scanning. These datasets were deliberately selected to balance diversity and manageability, ensuring robust model development while avoiding potential challenges associated with too many datasets, such as increased computational complexity, overfitting due to redundancy, and resource overhead. Both datasets feature 43 common attributes that describe various aspects of network traffic, such as connection duration, protocols, packet size, and the number of packets transmitted. These shared features enable effective analysis and model development for detecting intrusions, particularly DDoS attacks in IoT environments.

The following tables show the class distribution of each dataset:

TABLE I
NF-BoT-IoT-V2 Classes

Classes	Value
Benign	443
DDoS	51140
DoS	48417

The NF-BoT-IoT-V2 dataset, as shown in Table I, includes three primary classes: Benign (443 instances), DDoS (51,140 instances), and DoS (48,417 instances). These classes represent regular network traffic and different types of denial-of-service attacks, with the DDoS and DoS categories being the most prevalent.

TABLE II
NF-CSE-CIC-IDS2018-V2 Classes

Classes	Value
Benign	89463
DDoS attacks-LOIC-HTTP	6362
DDOS attack-HOIC	2630
DoS attacks-Slowloris	1226
DoS attacks-Hulk	163
DoS attacks-GoldenEye	88
DoS attacks-SlowHTTPTest	59
DDOS attack-LOIC-UDP	9

The NF-CSE-CIC-IDS2018-V2 dataset, as shown in Table II, includes various attack classes, such as Benign (89,463 instances) and several DDoS and DoS variants, like DDoS attacks-LOIC-HTTP (6,362 instances) and DoS attacks Slowloris (1,226 instances), among others. This dataset's diversity of attack types allows for a comprehensive evaluation of intrusion detection models across multiple attack scenarios.

In this study, the data preprocessing phase entailed the filtration of each dataset to a sample of 100,000 entries, thereby providing a representative subset for detecting DDoS attacks while maintaining a balance between performance and resource efficiency. The L4 SRC PORT label in the NFBoT-IoT-V2 dataset underwent a correction to an unsigned format, a modification implemented for consistency with the NF-CSE-CIC-IDS2018-V2 dataset. Furthermore, attack labels were converted to numeric values, a standard machine learning practice, to facilitate efficient computation and model training. Despite simplifying attack labels, key features that describe attack behaviors, such as packet size, duration, and protocols, were preserved, ensuring the model's ability to learn nuanced patterns effectively.

B. Task Selection

Task selection involves choosing specific tasks for the model to learn simultaneously. The selected tasks should be related in a way that helps predict the outcomes of other tasks. The tasks of this study are based on the type of attack from the NF-CSE-CIC-IDS2018-V2 dataset (Task 1) and the label from the NF-BoT-IoT-V2 dataset (Task 2). The tasks use supervised learning outputs provided for each dataset: Label and Attack. The Label task indicates whether an action is an attack. In contrast, the Attack task identifies the type of attack, such as benign, DDoS attacks-LOIC-HTTP, DDoS attacks-HOIC, DoS attacks-Slowloris, DoS attacks-Hulk, DoS attacks-GoldenEye, DoS attacks-SlowHTTPTest, and DDoS attacks-LOIC-UDP.

C. Deep Learning Model

Deep Learning is a machine learning method that utilizes multi-layered artificial neural networks to learn features and representations of patterns from data. The network has three layers: input, hidden, and output. The hidden layer contains neurons that are connected to the neurons in the previous and next layers [3]. For this study, three DL algorithms are applied: multi-layer perceptron (MLP), long-short-term memory (LSTM), and convolutional neural network (CNN) to evaluate their performance on each task.

1) MLP is a machine learning method that uses a multilayer artificial neural network to transfer information from the input layer through one or more hidden layers to the output layer. MLP employs supervised learning, which involves training the model with known target data and learning from the outcomes. The mathematical representation of MLP is as follows [12].

$$c = f(d) \text{ and } d = \sum_{i=0}^{n} (a_i b_i) \tag{1}$$

The process begins by assigning weights to each input, where each input b_i is multiplied by its weight a_i. Then, the result of these multiplications are summed to get d. Finally, the activation function is applied to d to yield the output c.

2) LSTM is a machine learning technique that employs a Recurrent Neural Network(RNN) to solve the problem of long-term data dependency. LSTM architecture comprises a series of cells, each with memory and three gates: forget, input, and output. These gates control the flow of information inflow into and outflow of the cells. One task that can be used with LSTM is traffic anomaly detection in computer networks [13]. This method guarantees effective information transmission and retention inside the LSTM network.

3) CNN was initially developed to process images but has proven effective for various data types, including sequential text. A CNN comprises convolution, pooling, and fully connected layers. The convolution layer extracts features from the input data, utilizing a specific activation function. The pooling layer then reduces the dimensionality of the data, employing techniques like max pooling to speed up processing and mitigate overfitting. Finally, the fully connected layer takes the flattened output from the pooling layer to generate the final output class [15]. The mathematical formula for a CNN can be expressed as follows:

$$W_l \in \mathbb{R}^{k_l \times k_l \times n_l \times n_{l-1}} \tag{2}$$

$$b_l \in \mathbb{R}^{n_l} \tag{3}$$

$$\{W_{\mathrm{fc},k}\} \in \mathbb{R}^{n_{\mathrm{fc},k-1} \times n_{\mathrm{fc},k}} \tag{4}$$

The convolution layer (2, 3) is defined by the weight W_l and bias b_l specific to each layer (l), he filter size is denoted by k_l, while n_l represent the number of filter layers, and n_{l-1} represent the number of channels of the input layer l. The pooling layer has no specific formula, but the type and size of pooling chosen determine its structure. The Fully Connected layer (4) is defined by the weights $\{W_{\mathrm{fc},k}\}$ and bias $\{b_{\mathrm{fc},k}\}$. Here, $n_{\mathrm{fc},k-1}$ represents the number of neurons in the previous fully connected layer, and $n_{\mathrm{fc},k}$ represents the number of neurons in the current fully connected layer [16].

D. Single-task Deep Learning Model

Single-task deep learning models are designed to tackle a specific task by focusing on one target variable. This study applies these models to two datasets: NF-CSECIC-IDS2018-V2 and NF-BoT-IoT-V2. The NF-CSE-CICIDS2018-V2 dataset aims to classify different types of attacks in network data. Therefore, the target variable is "Attack." Conversely, the NF-BoT-IoT-V2 data set is meant to detect normal and abnormal conditions in a network. Therefore, the target variable is "Label."

Three DL algorithms—MLP, LSTM, and CNN—were used. These algorithms were chosen because they can discover intricate patterns in network data.

The training process involved the following steps:

- Splitting both dataset data into 70% training data and 30% testing data.
- Preprocessing the data through normalization, encoding, and reshaping if required.
- Train models using the appropriate target for each dataset.
- Evaluating the models performance using the specified evaluation metrics.

The performance results of the three algorithms on each dataset are visualized in III and Table. IV

TABLE III
PREDICTION RESULT SINGLE TASK IN NF-CSE-CIC-IDS2018-V2

	Accuracy	Precision	Recall	F1-Score	Time (sec)
MLP	0.9975	0.9955	0.9975	0.9965	19.75
LSTM	0.9977	0.9957	0.9977	0.9967	230.69
CNN	0.9991	0.9991	0.9991	0.9990	27.11

Table III presents the performance of three MLP, LSTM, and CNN models on the NF-CSE-CIC-IDS2018-V2 dataset. The MLP model demonstrates high performance, achieving an accuracy of 0.9975 and a fast execution time of 19.75 seconds. The LSTM model slightly outperformed the MLP with an accuracy of 0.9977, though it requires a significantly longer execution time of 230.69 seconds. The CNN model has the highest accuracy of 0.9991, while maintaining a reasonable execution time of 27.11 seconds, making it the most efficient in accuracy and speed.

TABLE IV
PREDICTION RESULT SINGLE TASK IN NF-BOT-IOT-V2

	Accuracy	Precision	Recall	F1-Score	Time (sec)
MLP	0.9998	0.9998	0.9998	0.9998	18.76
LSTM	1.0	1.0	1.0	1.0	228.98
CNN	0.9999	0.9999	0.9999	0.9999	25.24

Table IV presents the results for the NF-BoT-IoT-V2 dataset. It shows that MLP achieves a nearly perfect accuracy of 0.9998 and has the fastest execution time of 18.76 seconds. While LSTM reaches a perfect accuracy of 1.0 across all metrics, it takes significantly longer to process at 228.98 seconds. On the other hand, CNN offers a very high accuracy of 0.9999 with a moderate execution time of 25.24 seconds, providing a good balance between speed and accuracy.

E. Multitask Deep Learning Model

In this research, we apply MTL combined with DL algorithms to detect and classify DDoS attacks. MTL allows the model to learn to identify the presence of attacks and their types simultaneously. This approach is expected to improve the model's accuracy by enhancing its ability to recognize patterns in the dataset. The algorithm for MTL is provided in [17].

$$\min_{\mathbf{W}=\{w_1,\ldots,w_m\}} \sum_{m=1}^{M} L(X^m, y^m, w^m) + \lambda \,\mathrm{Reg}(\mathbf{W}) \tag{5}$$

In this algorithm, $X^m \in \mathbb{R}^{N_m \times D}$ represents the input matrix for the m-th task, with N_m samples and D features, while $y^m \in \mathbb{R}^{N_m \times 1}$ is the corresponding output vector. The task-specific weights, denoted as $w^m \in \mathbb{R}^{D \times 1}$, are learned for each task. The global weight matrix $\mathbf{W} = [w_1, w_2, \ldots, w_M]$ is formed by concatenating the individual task-specific weights. The regularization term λ balances the loss from the tasks and prevents overfitting. This helps ensure the model can generalize well across all tasks. We developed three models to implement MTL for attack detection and classification, each utilizing various DL architectures designed for MTL.

1) Model Construction: Three Multitask DL models were constructed:

- **MTL-MLP:** A multi-layer perceptron with two dense layers (32, 16 neurons) and ReLU activation.
- **MTL-LSTM:** Two LSTM layers (32, 16 units) with dropout layers for better generalization.
- **MTL-CNN:** A convolutional layer with 32 filters, followed by pooling layers to reduce dimensionality.

Initially, the two datasets are split into 70% for training and 30% for testing. the target variables in the models are `one-hot encoded` , while features are normalized with `StandardScaler`. Then, input data is reshaped to fit the specified model requirements, such as sequences for LSTM models or 2D formats for CNNs. Finally, the `Adam` optimizer are used to optimize the models which are trained over five epochs.

2) Evaluation: The models were evaluated based on *accuracy*, *precision*, *recall*, *F1-score*, and the *execution time(sec)*.

IV. RESULT AND DISCUSSION

This section presents the final results of our research and offers recommendations for future work.

A. Experiment Result

The experiment results in Fig. 2 show that the performance of MLP, LSTM, and CNN models in NF-CSE-CIC-IDS2018-V2 and NF-BoT-IoT-V2 datasets can be improved effectively by MTL. In this study, MTL improved the accuracy of the STL MLP model in Task 1 from 0.9975 to 0.9984, precision from 0.9955 to 0.9984, recall from 0.9975 to 0.9984, and F1-score from 0.9965 to 0.9978, while slightly increasing execution time from 19.75 seconds to 22.48 seconds. While, for the LSTM model in Task 1 and Task 2, getting slightly lower across all metric compared to STL and execution time increased from 230.69 seconds to 311.39 seconds. The CNN model's scores improved across all metrics for the Task 1 and perfect score for the Task 2, with the execution time growing from 27.11 seconds to 32.77 seconds.

The increased execution time in MTL arises because the model processes both datasets simultaneously, which requires more computational resources than STL. Although processing each dataset sequentially would take longer, combining them in MTL results in a more efficient model. This approach leverages shared features across datasets, leading to better overall performance despite the additional time needed for training.

As demonstrated, the confusion matrix in fig 3, the MTL-MLP model demonstrates optimal performance in multiclass classification and nearly perfect accuracy, indicating its capacity to discern complex patterns. Simultaneously, the MTL-CNN model attains perfect results in binary classification , substantiating its superiority in recognizing specific patterns in binary data. It is important to note that, despite the inherent data imbalance, no balancing techniques were applied during this research. The decision to retain the original data distribution was made because the models already demonstrated exceptional performance, highlighting their robustness in handling imbalanced data effectively without compromising accuracy. This approach emphasizes the sufficiency of the current dataset and the compatibility of the selected model architectures with the task complexities.

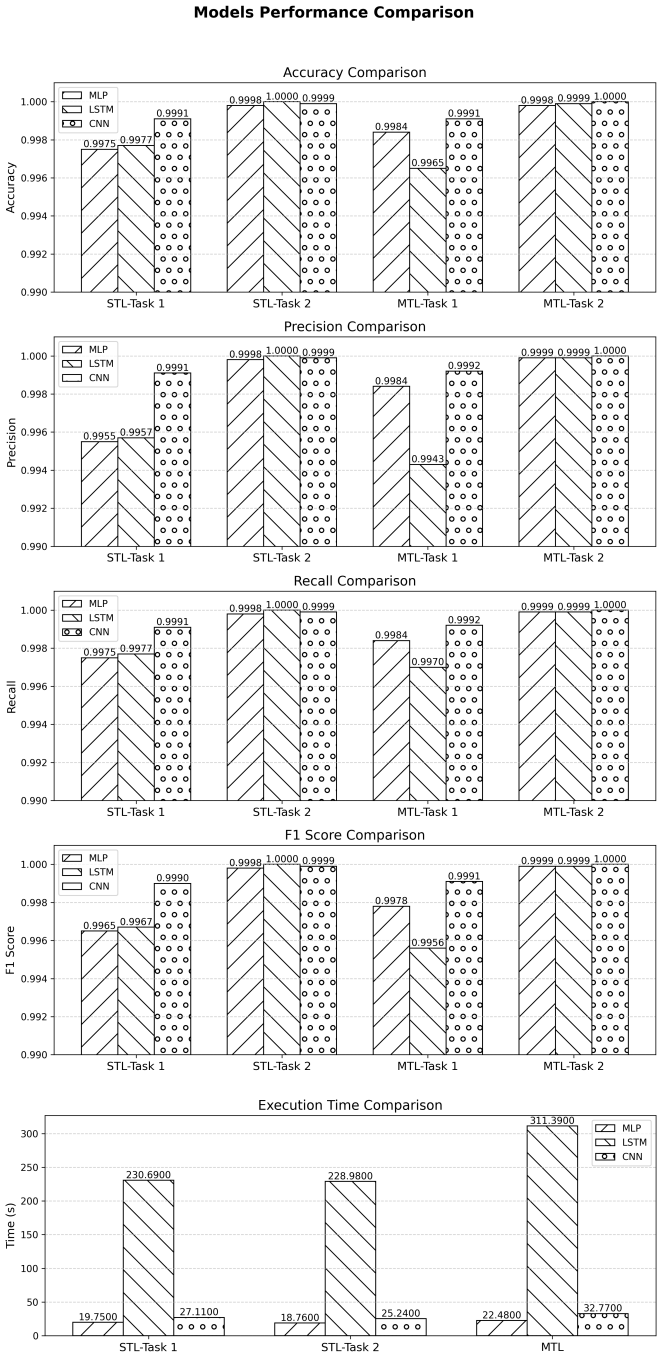

Fig. 2. MTL and DL Algorithm Result

B. Discussion

MTL requires more execution time than STL. However, MTL's significant improvements in accuracy, precision, recall,

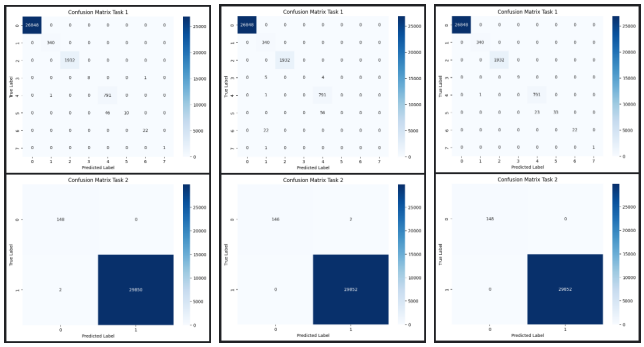

Fig. 3. CM MTL-MLP, CM MTL-LSTM, and CM MTL-CNN

and F1-score demonstrate its effectiveness for DDoS attack detection. The additional time MTL requires is due to its complex optimization processes, which leverage inter-task relationships to enhance performance. For instance, on the NF-CSE-CIC-IDS2018-V2 dataset using multiclass classification, MTL-Task 1 enhances precision from 0.9955 (STL MLP) to 0.9992 (MTL-CNN), demonstrating its capacity to utilize shared patterns effectively. Similarly, on the NF-BoT-IoT-V2 dataset using binary classification, MTL-Task 2 attains perfect precision, recall, and F1-score with MTL-MLP, MTL-LSTM and MTL-CNN models, demonstrating its capacity to generalize effectively on less complex datasets. Despite the increased execution times, MTL's enhanced accuracy and generalization make it a compelling choice for complex tasks such as DDoS attack detection.

While real-time detection and traditional methods, such as statistical or rule-based approaches, are crucial for broader applicability, the primary focus of this study was to evaluate the benefits of MTL in enhancing DDoS detection. The increase in execution time for LSTM with MTL, though noteworthy, is outside the scope of this research, which does not extend to real-time detection constraints. Additionally, the comparison to traditional methods was not included, as the research was centered on DL-based models and their potential for advancing intrusion detection performance. However, these aspects provide valuable directions for future work, including optimization for real-time performance and comparisons with traditional detection methods.

Conclusion

The research highlights the significant potential of advanced machine learning techniques in enhancing Network Intrusion Detection Systems (NIDS). The assessment of various algorithms such as MLP, LSTM, and CNN on datasets like NF-CSE-CIC-IDS2018-V2 and NF-BoT-IoT-V2 indicates that CNN achieves the highest performance. It shows nearly perfect accuracy, precision, recall, and F1-score while providing efficient prediction times. Additionally, incorporating NetFlow-based features, as demonstrated in Sarhan et al.'s work, consistently improves classification performance across different datasets, emphasizing the importance of standardized feature sets for more reliable and comparable evaluations.

References

[1] K. B. Adedeji, A. M. Abu-Mahfouz, and A. M. Kurien, "DDoS Attack and Detection Methods in Internet-Enabled Networks: Concept, Research Perspectives, and Challenges," *Journal of Sensor and Actuator Networks*, vol. 12, no. 4, p. 51, July 2023, doi: 10.3390/jsan12040051.

[2] M. Crawshaw, "Multi-Task Learning with Deep Neural Networks: A Survey," *arXiv*, September 2020. [Online]. Available: http://arxiv.org/abs/2009.09796. [Accessed: May 15, 2024].

[3] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS Attack via Deep Learning," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, Hong Kong, China, May 2017, pp. 1–8, doi: 10.1109/SMARTCOMP.2017.7946998.

[4] A. A. Wardana, P. Sukarno, S. Basuki, and S. B. Utomo, "Federated Random Forest with Feature Selection for Collaborative Intrusion Detection in Internet of Things," in *Procedia Computer Science*, vol. 246, pp. 20-29, 2024. DOI: 10.1016/j.procs.2024.09.193.

[5] A. A. Wardana and P. Sukarno, "Taxonomy and Survey of Collaborative Intrusion Detection System using Federated Learning," *ACM Computing Surveys*, vol. 57, no. 4, April 2025. DOI: 10.1145/3701724.

[6] Saleh A. Albelwi, "An Intrusion Detection System for Identifying Simultaneous Attacks using Multi-Task Learning and Deep Learning," in *2022 2nd International Conference on Computing and Information Technology (ICCIT)*, Jan. 25 - 27, 2022, Tabuk, Saudi Arabia. IEEE, 2022, pp. 349-353. DOI: 10.1109/ICCIT52419.2022.9711630.

[7] S. Shakya and R. Abbas, "A Comparative Analysis of Machine Learning Models for DDoS Detection in IoT Networks," *arXiv*, November 2024. [Online]. Available: http://arxiv.org/abs/2411.05890. [Accessed: November 26, 2024].

[8] J. Halladay, D. Cullen, N. Briner, J. Warren, K. Fye, R. Basnet, J. Bergen, and T. Doleck, "Detection and Characterization of DDoS Attacks Using Time-Based Features," *IEEE Access*, vol. 10, pp. 1–1, Jan. 2022, doi: 10.1109/ACCESS.2022.3173319.

[9] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a Standard Feature Set for Network Intrusion Detection System Datasets," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 357–370, February 2022, doi: 10.1007/s11036-021-01843-0.

[10] S. Ali, O. Abusabha, F. Ali, M. Imran, and T. Abuhmed, "Effective Multitask Deep Learning for IoT Malware Detection and Identification Using Behavioral Traffic Analysis," in *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1199-1209, June 2023. DOI: 10.1109/TNSM.2022.3200741.

[11] Q. Liu, D. Wang, Y. Jia, S. Luo, and C. Wang, "A multi-task based deep learning approach for intrusion detection," in *Knowledge-Based Systems*, vol. 238, 2022, Art. no. 107852. DOI: 10.1016/j.knosys.2021.107852.

[12] H. Taud and J. F. Mas, "Multilayer Perceptron (MLP)," in *Geomatic Approaches for Modeling Land Change Scenarios*, M. T. Camacho Olmedo, M. Paegelow, J.-F. Mas, and F. Escobar, Eds., Lecture Notes in Geoinformation and Cartography, Springer International Publishing, 2018, pp. 451–455, doi: 10.1007/978-3-319-60801-3_27.

[13] H. A. Gouda, M. A. Ahmed, and M. I. Roushdy, "Optimizing anomaly-based attack detection using classification machine learning," *Neural Computing and Applications*, vol. 36, no. 6, pp. 3239–3257, February 2024, doi: 10.1007/s00521-023-09309-y.

[14] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, July 2019, doi: 10.1162/neco_a_01199.

[15] A. R. Shaaban, E. Abd-Elwanis, and M. Hussein, "DDoS attack detection and classification via Convolutional Neural Network (CNN)," in *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, Cairo, Egypt, December 2019, pp. 233–238, doi: 10.1109/ICICIS46948.2019.9014826.

[16] G. Carneiro, J. Nascimento, and A. P. Bradley, "Deep Learning Models for Classifying Mammogram Exams Containing Unregistered Multi-View Images and Segmentation Maps of Lesions," in *Deep Learning for Medical Image Analysis*, Elsevier, 2017, pp. 321–339, doi: 10.1016/b878-0-12-810408-8.00019-5.

[17] Thung, K. H., and Wee, C. Y. (2018). A brief review on multi-task learning. *Multimedia Tools and Applications*, 77, 29705–29725. doi: 10.1007/s11042-018-6463-x