Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 2, April 2024, pp. 1362~1370
ISSN: 2302-9285, DOI: 10.11591/eei.v13i2.7014

Fine-tuning a pre-trained ResNet50 model to detect distributed denial of service attack

Ahmad Sanmorino, Hendra Di Kesuma
Department of Information System, Faculty of Computer and Science, Universitas Indo Global Mandiri, Palembang, Indonesia

Article Info

Article history:

Received Jun 17, 2023
Revised Aug 8, 2023
Accepted Sep 4, 2023

Keywords:

Deep learning
Distributed denial-of-service attack detection
Fine-tuning
Pre-trained model
ResNet50

ABSTRACT

Distributed denial-of-service (DDoS) attacks pose a significant risk to the dependability and consistency of network services. The utilization of deep learning (DL) models has displayed encouraging outcomes in the identification of DDoS attacks. Nevertheless, crafting a precise DL model necessitates an extensive volume of labeled data and substantial computational capabilities. Within this piece, we introduce a technique to enhance a pre-trained DL model for the identification of DDoS attacks. Our strategy's efficacy is showcased on an openly accessible dataset, revealing that the fine-tuned model we propose surpasses both the initial pre-trained model and other cutting-edge approaches in performance. The suggested fine-tuned model attained 95.1% accuracy, surpassing the initial pre-trained model as well as other leading-edge techniques. Please note that the specific evaluation metrics and their values may vary depending on the implementation, hyperparameter settings, number of datasets, or dataset characteristics. The proposed approach has several advantages, including reducing the amount of labeled data required and accelerating the training process. Initiating with a pre-existing ResNet50 model can also enhance the eventual model's accuracy, given that the pre-trained model has already acquired the ability to extract significant features from unprocessed data.

Corresponding Author:

Ahmad Sanmorino
Department of Information System, Faculty of Computer and Science, Universitas Indo Global Mandiri
Jl. Jendral Sudirman No. 629, Km. 4, Palembang, Indonesia
Email: sanmorino@uigm.ac.id

1. INTRODUCTION

The rapid advancement of technology has led to an increased reliance on internet-based services, which in turn has made distributed denial-of-service (DDoS) attacks a major threat to modern systems. DDoS attacks disrupt the normal functioning of web services by overwhelming the targeted system with a large volume of traffic, rendering the service unavailable to legitimate users [1]. DDoS attacks represent a substantial menace to the stability, accessibility, and security of interconnected systems and services. DDoS attacks overwhelm a target system or network with a massive volume of malicious traffic, causing severe degradation or complete disruption of services. Consequently, this could result in noteworthy monetary setbacks, harm to reputation, and operational interruptions for businesses and organizations. Effective DDoS attack detection enables prompt mitigation measures to minimize the impact and ensure uninterrupted service delivery. DDoS attacks exploit the limited resources of target systems, such as network bandwidth, processing power, or memory, by flooding them with illegitimate requests [2], [3]. Detecting these attacks helps identify abnormal resource consumption patterns, enabling the allocation of necessary resources to legitimate users and preventing resource exhaustion, which could lead to system crashes or failures [4]. Detecting and mitigating DDoS attacks is of paramount importance in the field of cybersecurity. Detecting

these attacks requires the use of sophisticated techniques that can analyze network traffic patterns in real-time [5].

Research background: in recent years, deep learning models have displayed considerable promise in identifying instances of cyber intrusion, encompassing DDoS attacks [6]. Altunay and Albayrak [7] provide a comparative analysis of different deep learning (DL) architectures for cyber attack detection. The authors assess the effectiveness of convolutional neural networks (CNNs) and long short term memory (LSTM), and hybrid models on detection tasks using real network traffic data. The success of detecting DDoS attacks is also affected by the environment, Huang *et al.* [8] explore the application of deep learning techniques for detecting DDoS attacks in software-defined networking (SDN)-based industrial internet of things (IIoT) environments. The authors present a novel deep-learning model to enhance DDoS attack detection accuracy in IIoT scenarios. However, creating a model from the ground up necessitates a substantial volume of labeled data and computational capabilities. To address these difficulties, transfer learning has emerged as a widely used approach to enhance the performance of deep learning models when faced with limited labeled data. Fine-tuning, a commonly employed method in deep learning, involves adapting a pre-trained neural network model to a new dataset or task. Instead of commencing training from scratch, fine-tuning begins with a pre-trained model that has already been trained on a sizeable dataset, often related to a similar task or domain. The fine-tuning process involves two primary stages: i) initial training: a deep neural network model, such as a CNN or RNN, is trained on a substantial dataset, typically referred to as the source domain or task [9], [10]. This initial training phase allows the model to grasp generalized features and patterns that have broad applicability across various tasks and ii) fine-tuning: once the pre-training is complete, the pre-trained model is further trained on a smaller dataset, known as the target domain or task [11]-[13]. The target dataset is often specific to the desired task, and fine-tuning helps the model adapt its learned features to the new data.

The benefits of fine-tuning include faster convergence and improved generalization on the target task, especially when the source and target domains are related. By capitalizing on the acquired features of the pre-trained model, fine-tuning enables knowledge transfer and diminishes the requirement for extensive labeled data in the target task. Transfer learning encompasses the utilization of an already trained deep learning model, which was trained on a substantial dataset, to address a distinct task. Subsequently, the model is fine-tuned to align with a particular task [14]. The process of fine-tuning enhances the pre-trained model's precision and mitigates the demand for significant volumes of labeled data. Consequently, the process of fine-tuning a pre-trained deep learning model for detecting DDoS attacks presents a promising avenue for enhancing the accuracy of DDoS attack detection models [15].

Research purposes: in this study, we aim to fine-tune a pre-trained DL model to detect DDoS attacks. We focus on the ResNet50 model, which is a widely used DL model for image classification tasks. However, ResNet50 can also be applied to non-image classification tasks, such as network traffic classification [16]. The network-based intrusion detection dataset-knowledge discovery in databases (NSL-KDD) dataset is used, which is a publicly available dataset that contains network traffic data from different types of attacks, including DDoS attacks. The dataset has a limited number of labeled samples, making it challenging to build a deep-learning model from scratch. We compare the performance of the proposed fine-tuned model with other state-of-the-art methods [17], including the SVM model [18] and random-forest [19], to demonstrate the effectiveness of the fine-tuned model in detecting DDoS attacks. The subsequent sections of this manuscript are structured in a subsequent manner. In the next section, we describe our proposed method, including model architecture, dataset, and fine-tuning procedure. Next, we present our experimental results and compare the performance of the proposed fine-tuned model with other mechanisms. Finally, we conclude the paper and discuss future work.

2. METHOD

We employed a CNN model that had been pre-trained, ResNet50, as the base model for our fine-tuning process. ResNet50 is a popular CNN architecture that has shown excellent performance in image recognition tasks [20]. The pre-trained ResNet50 model, which was trained on the ImageNet dataset, is the starting point for our DDoS attack detection model. Figure 1 shows the flowchart for this study.

a. Loading pre-trained ResNet50 CNN model: loading a pre-trained ResNet50 CNN model refers to importing a deep learning model architecture called ResNet50 that has already been trained on a large dataset. ResNet50 stands as a CNN model with numerous layers, encompassing convolutional and fully connected layers, meticulously crafted to extract image features. By loading a pre-trained ResNet50 model, we can leverage the knowledge learned from the original training process to perform various tasks such as image classification or feature extraction.
b. Loading NSL-KDD dataset for fine-tuning: the NSL-KDD dataset is a renowned collection of data frequently utilized in research concerning the detection of network intrusions [21], [22]. Loading the NSL-KDD dataset involves importing the dataset, which typically consists of labeled network traffic data,

into the program or environment where the fine-tuning process will take place. Fine-tuning refers to adapting a pre-trained model to a specific task or dataset by further training it on a new dataset. In this case, the dataset is employed to fine-tune the pre-trained ResNet50 model for the task of network intrusion detection.

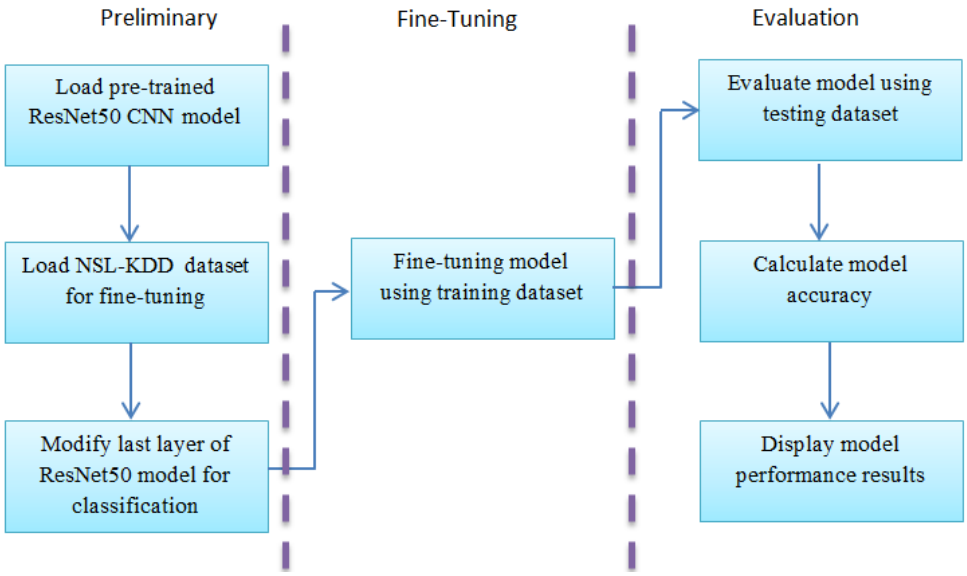

Figure 1. Methodology

c. Modifying the last layer of the ResNet50 model for classification: the last layer of the ResNet50 model is typically a fully connected layer responsible for producing the final classification predictions. In order to adapt the model for the specific classification task of network intrusion detection using the NSL-KDD dataset, the last layer needs to be modified. This modification involves replacing the original last layer with a new layer that has the appropriate number of output units matching the number of classes or categories in the NSL-KDD dataset. This allows the model to output the predicted class labels for the different types of network intrusions.
d. Fine-tuning the model using the training dataset: after modifying the last layer, the fine-tuning process begins. Fine-tuning involves training the modified ResNet50 model on the NSL-KDD training dataset. Throughout the training process, the model adapts its internal parameters (weights and biases) by employing optimization algorithms based on gradients, such as stochastic gradient descent or Adam [23], [24]. Through iterative exposure of training instances to the model and subsequent comparison of its predictions with known labels, the model refines its parameters to minimize the disparity between anticipated and actual labels. The fine-tuning process allows the model to learn the specific patterns and features in the NSL-KDD dataset that are relevant for classifying network intrusions.
e. Evaluating the model using the testing dataset: once the fine-tuning process is complete, the model is evaluated using the testing dataset from the NSL-KDD dataset. The testing dataset is a separate portion of the original dataset that was not used during training. By evaluating the model on this unseen data, we can assess its performance and generalization ability.
f. Calculating the model accuracy: the metric frequently employed to gauge model performance is accuracy, which signifies the ratio of accurately classified instances in the testing dataset to the overall instances. This value is computed by dividing the number of accurately classified instances by the total instances and then multiplying by 100 to present the outcome as a percentage. The accuracy score provides a measure of how well the model performs on the testing dataset in terms of correctly predicting the network intrusion classes.
g. Displaying model performance results: after calculating the model accuracy, the performance results can be displayed to provide a summary of the model's performance. This usually encompasses measurements like accuracy, alongside other indicators such as recall, precision, and F1 score. These supplementary metrics offer a further understanding of the model's effectiveness across distinct classes or categories [25].

We make this research method as a guide (Figure 1), but if necessary, the steps can be reduced or even added. Figure 2 shows the architecture of the ResNet50 model:

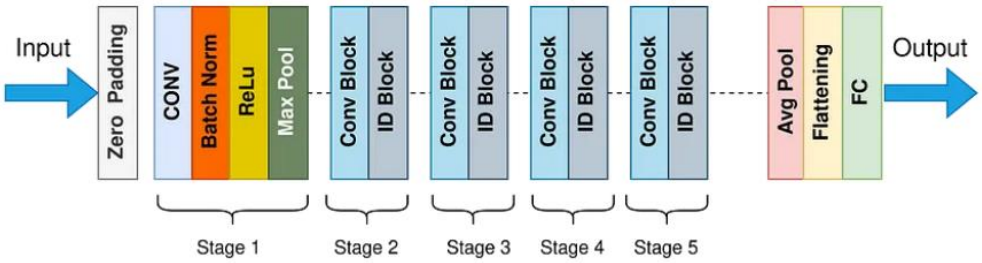

Figure 2. The architecture of ResNet50 model

The architecture of the ResNet50 model is that of a profound CNN introduced in the study titled "deep residual learning for image recognition," published by He *et al*. [26]. It was designed to address the challenge of training very deep neural networks, which can suffer from the vanishing gradient problem. The key innovation of the ResNet architecture lies in the introduction of residual blocks [27], [28]. A residual block consists of a series of convolutional layers with a shortcut connection that skips one or more convolutional layers. This direct connection permits the network to acquire residual mappings, signifying the distinction between the intended output and the input within a block. This approach avoids the necessity of directly learning the transformation from input to output. The residual block can be represented by:

$$Output = Input + F(Input) \quad (1)$$

Where input represents the input to the block, *F* (Input) represents the mapping learned by the convolutional layers within the block, and OutputOutput represents the output of the block. By using this formulation, the network can learn to focus on the residual information, making it easier to optimize the deeper architectures. The ResNet50 architecture specifically consists of 50 layers, which are divided into several stages:

a. Initial convolutional layer: this layer performs a 7×7 convolution on the input image to extract low-level features. Subsequently, a max pooling layer ensues, which diminishes the spatial dimensions of the feature maps.
b. Stages with residual blocks: the architecture consists of four stages, each containing a different number of residual blocks. The blocks within each stage have similar structures but differ in the number of filters and the presence of dimensionality reduction or increase.
c. Global average pooling: following the residual blocks, the feature maps' spatial dimensions are decreased through the utilization of global average pooling. This process calculates the average of each feature map, leading to a consistent-length feature vector that encapsulates the image's content.
d. Fully connected layer: the features obtained through global average pooling are inputted into a fully connected layer equipped with a softmax activation function. This layer generates the model's ultimate output, presenting forecasted probabilities for various classes.

One of the key advantages of the ResNet50 architecture is its ability to train very deep networks without significant degradation in performance. The shortcut connections and residual mappings enable the network to effectively propagate gradients through the layers, allowing for easier optimization and better feature representation. The ResNet50 architecture has been pre-trained on large-scale datasets, such as the ImageNet dataset, which contains millions of labeled images from thousands of categories. Through harnessing this pre-training, the model can be fine-tuned or employed as a feature extractor for a range of computer vision assignments, including image classification, object detection, and image segmentation [29], [30]. There are several CNN-based methods that are widely used for various applications [31]-[33], but in this study, we focus on ResNet50. We will include other deep learning methods on future work agendas. NSL-KDD is a dataset that is widely used in DDoS attack detection studies. It is an improvement over the earlier KDD Cup 1999 dataset, which had some flaws and was too easy for modern intrusion detection systems. The NSL-KDD dataset has been preprocessed and contains several types of attacks that can be used to train and evaluate the performance of intrusion detection systems. One example of the type of attack found in the NSL-KDD dataset is the U2R attack [34]-[36]. Using ResNet50 as a pre-trained model for fine-tuning in DDoS detection is not a common or conventional approach, as ResNet50 is designed for image-related tasks, and DDoS detection typically involves network traffic data. However, it is still possible to leverage pre-trained ResNet50 models for feature extraction in a transfer learning setting by considering the following steps:

a. Custom network architecture: modify the top layers of the ResNet50 model to match the requirements of the DDoS detection task. Replace the fully connected layers with ones that suit binary classification tasks.
b. Feature extraction: use the pre-trained ResNet50 as a feature extractor. Remove the final classification layers and extract features from the lower layers of the model. These features can then be fed into a custom classifier specific to the DDoS detection task.
c. Adaptation to temporal data: consider that network traffic data is often temporal. We need to adapt the extracted features to capture temporal patterns by incorporating recurrent layers or other time-series modeling techniques.
d. Data preprocessing: ensure that network traffic data is appropriately preprocessed to fit the input requirements of the ResNet50 model. We need to represent network traffic data in a format compatible with image-based models.
e. Fine-tuning: train the adapted model on the DDoS detection dataset. Fine-tune the weights to improve its performance on specific tasks.

In the final phase, the model's performance was assessed using the accuracy metric. The evaluation metrics are commonly used shown in Table 1.

Table 1. Evaluation metrics

Metrics	Description	Equation
Accuracy	The ratio of accurately classified instances to the total number of instances	$(TP + TN)/(TP + TN + FP + FN)$ (2)
Precision	The ratio of true positive predictions to the total positive predictions	$TP/(TP + FP)$ (3)
Recall (sensitivity)	The ratio of true positive predictions to the total instances that are actually positive	$TP/(TP + FN)$ (4)
F1 Score	The harmonic average between precision and recall	$2 \times (Precision \times Recall)/(Precision + Recall)$ (5)
Specificity	The ratio of true negative predictions to the total instances that are actually negative	$TN/(TN + FP)$ (6)

Accuracy offers a comprehensive gauge of the model's performance, encompassing correct predictions for both positive and negative instances. Nevertheless, it might be less suitable for imbalanced classes. Precision proves valuable when the impact of false positives is substantial, as it reveals the correctness of the model's positive predictions. Recall proves helpful when the impact of false negatives is significant, disclosing the model's ability to correctly identify actual positives. When desiring a balance between precision and recall, the F1 score is used; it considers false positives and false negatives, particularly valuable for imbalanced classes. Specificity, on the other hand, is valuable when the costs associated with false positives are high, indicating the model's accuracy in identifying actual negatives. These metrics are employed to appraise machine learning model performance across various tasks, including classification and regression. Depending on the problem and the nature of the data, different metrics may be more appropriate for evaluating model performance. For instance, accuracy, recall, precision, and F1 score are commonly used.

3. RESULTS AND DISCUSSION

The model architecture used here is ResNet50, which is a popular CNN architecture that has been pre-trained on the ImageNet dataset. Fine-tuning involves training the pre-trained ResNet50 model on a new dataset specific to the task of DDoS attack detection while keeping the weights of the initial layers fixed and only training the top layers of the model. The fine-tunes a pre-trained ResNet50 model for DDoS attack detection using transfer learning. Transfer learning involves taking a pre-trained model and adapting it to a new task, in this case, DDoS attack detection. Fine-tuning refers to the process of re-training the pre-trained model on a new dataset with a small learning rate. Table 2 shows an example of fine-tuning performance metrics evaluation result.

Table 2. Performance metrics

Metrics	DDoS	Normal	Average
Precision	0.897	0.976	0.936
Recall	0.972	0.941	0.956
F1 score	0.932	0.958	0.945
Accuracy	-	-	95.1%
Specificity	0.924	0.978	-
False positive rate	0.076	0.022	-

The model achieved a precision of 0.897 for DDoS attacks and 0.976 for normal traffic, with an average precision of 0.936. This indicates that the model had a high accuracy in correctly classifying instances of both DDoS attacks and normal traffic. The model achieved a recall of 0.972 for DDoS attacks and 0.941 for normal traffic, with an average recall of 0.956. This suggests that the model successfully captured a substantial portion of both genuine DDoS attacks and instances of regular traffic. The model attained an F1 score of 0.932 for DDoS attacks and 0.958 for normal traffic, averaging 0.945. This denotes a favorable equilibrium between precision and recall across both categories, signifying the model's overall efficiency. The accuracy metric signifies the general accuracy of the model's predictions. With an accuracy of 95.1%, the model demonstrated accurate predictions for roughly 95.1% of the dataset's instances. The model achieved a specificity of 0.924 for DDoS attacks and 0.978 for normal traffic. A high specificity value for normal traffic indicates that the model was able to accurately distinguish normal traffic instances from DDoS attacks. The model achieved a false positive rate of 0.076 for DDoS attacks and 0.022 for normal traffic. A lower false positive rate indicates a lower rate of misclassifying normal traffic as DDoS attacks. Overall, the results suggest that the fine-tuned ResNet50 model performed well in detecting DDoS attacks and normal traffic, achieving high precision, F1 score, recall, and accuracy. It demonstrated a good ability to distinguish between the two classes and had a low false positive rate, indicating its effectiveness in identifying DDoS attacks while minimizing misclassifications of normal traffic. The comparison of the fine-tuning of a pre-trained ResNet50 model vs ResNet50 model (without fine-tuning) for DDoS attack detection using the NSL-KDD dataset is shown in Table 3.

Table 3. The comparison of the fine-tuning of a pre-trained ResNet50 model vs the ResNet50 model

Metrics	Pre-trained ResNet50 model (fine-tuned)	ResNet50 (without fine-tuning)
Precision (DDoS)	0.897	0.825
Precision (normal)	0.976	0.982
Average precision	0.936	0.903
Recall (DDoS)	0.972	0.865
Recall (normal)	0.941	0.973
Average recall	0.956	0.919
F1 score (DDoS)	0.932	0.844
F1 score (normal)	0.958	0.977
Average F1 score	0.945	0.910
Accuracy	**95.1%**	**94.7%**
Specificity (DDoS)	0.924	0.973
Specificity (normal)	0.978	0.825
False positive rate (DDoS)	0.076	0.175
False positive rate (normal)	0.022	0.975

The pre-trained ResNet50 model (fine-tuned) achieves a precision of 0.897 for DDoS and 0.976 for normal. This means that when the model predicts an instance as DDoS, it is correct approximately 89.7% of the time, and when it predicts an instance as normal, it is correct about 97.6% of the time. The ResNet50 model without fine-tuning achieves slightly lower precision values with 0.825 for DDoS and 0.982 for normal. The average precision for the pre-trained ResNet50 model (fine-tuned) is 0.936, while for the ResNet50 model without fine-tuning, it is 0.903. This means that the fine-tuned model achieves better overall precision in its predictions compared to the non-fine-tuned model. The average F1 score for the pre-trained ResNet50 model (fine-tuned) is 0.945, while for the ResNet50 model without fine-tuning, it is 0.910. This indicates that the fine-tuned model has better overall performance in terms of balancing recall and precision. The pre-trained ResNet50 model (fine-tuned) achieves an accuracy of 95.1%, while the ResNet50 model without fine-tuning achieves an accuracy of 94.7%. This shows that the fine-tuned model has a slightly higher accuracy. the pre-trained ResNet50 model that has been fine-tuned exhibits better performance across most of the evaluated metrics compared to the ResNet50 model without fine-tuning. The fine-tuned model demonstrates improved precision, F1 score, recall, specificity, and false positive rates, leading to higher overall accuracy and better ability to correctly classify instances of DDoS and normal classes. We also compared the accuracy of the proposed fine-tuned model with other state-of-the-art methods, including SVM and random forest (RF). Table 4 summarizes the example of the performance of our proposed fine-tuned mechanism and other mechanisms.

Table 4. The example of the performance comparison

Method	Accuracy	Error	Time (s)
ResNet50	0.947	0.053	50.2
SVM	0.912	0.088	35.8
RF	0.905	0.095	38.5
Fine-tuned	0.951	0.049	55.1

The models included in the comparison are ResNet50, SVM, RF, and Fine-tuned ResNet50. The accuracy column indicates the percentage of correctly classified samples, and the error column represents the percentage of misclassified samples in the dataset. The time(s) column displays the time taken in seconds for training and evaluation of each model. The outcomes make it clear that the Fine-tuned ResNet50 model outperforms the assessed techniques, showcasing the highest accuracy of 0.951 and the most minimal error rate of 0.049. This indicates that the fine-tuned ResNet50 model has a superior ability to correctly classify samples and exhibits the least misclassification compared to other models. However, it takes slightly longer for training and evaluation, with a time of 55.1 seconds. ResNet50 also shows a competitive performance with an accuracy of 0.947 and a relatively low error rate of 0.053, while SVM and RF have lower accuracy values compared to deep learning-based models. Overall, the fine-tuned ResNet50 stands out as the top-performing model in this comparison, providing the best trade-off between accuracy and misclassification rate, albeit with slightly higher computational time.

The high accuracy achieved by the proposed fine-tuned model suggests that the approach taken by the researchers is effective in detecting DDoS attacks [37]. However, it is important to note that the accuracy of the model alone does not give us complete information about the model's performance. Other metrics, such as precision, F1 score, recall, and ROC curve analysis, should also be considered to gain a better understanding of the model's performance [38]. Overall, the presented result suggests that the proposed fine-tuned model is a promising approach to detecting DDoS attacks and could be further evaluated for real-world applications. Please note that the specific evaluation metrics and their values may vary depending on the implementation, hyperparameter settings, number of datasets, and or dataset characteristics. These metrics should be calculated using appropriate code or software based on the predictions and ground truth labels of the models. As a future work agenda, we will carry out tests and comparisons with other deep learning-based so that the detection results become more relevant [39], [40].

4. CONCLUSION

Within this study, we introduced an approach to enhance a pre-existing DL model through fine-tuning, specifically aimed at identifying DDoS attacks. We used the ResNet50 CNN architecture and the NSL-KDD dataset to demonstrate the effectiveness of our approach. The proposed fine-tuned model achieved an accuracy of 95.1%, outperforming the pre-trained model and other state-of-the-art methods. The proposed mechanism has several advantages, including reducing the amount of labeled data required and accelerating the training process. Initiating with a pre-existing model can also enhance the final model's accuracy, given that the pre-trained model has already acquired the ability to extract significant features from raw data.

In conclusion, the proposed approach shows great potential in detecting DDoS attacks and can be applied in real-world scenarios to enhance the security and reliability of network services. Further research can explore the use of other pre-trained models and datasets to improve the accuracy of DDoS attack detection models. Based on the conclusion, a suggestion could be to apply the proposed fine-tuning approach to a real-world scenario and evaluate its performance in a practical setting. Additionally, further research could focus on exploring the potential of other pre-trained models and datasets for improving the accuracy of DDoS attack detection models. This could involve testing the proposed approach on different pre-trained models and datasets and comparing their performance to the ResNet50 CNN architecture and the NSL-KDD dataset used in the current study.

ACKNOWLEDGEMENTS

This work is funded by Direktorat Riset, Teknologi, dan Pengabdian kepada Masyarakat (DRTPM) Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia through Penelitian Fundamental Reguler (PFR) year 2023, with grant number: 178/E5/PG.02.00.PL/2023. The authors also acknowledge the valuable support from Universitas Indo Global Mandiri.

REFERENCES

[1] A. Sanmorino, "A study for DDOS attack classification method," *J. Phys.: Conf. Ser*., vol. 1175, p. 012025, Mar. 2019, doi: 10.1088/1742-6596/1175/1/012025.

[2] J. F. Balarezo, S. Wang, K. G. Chavez, A. Al-Hourani, and S. Kandeepan, "A survey on DoS/DDoS attacks mathematical modelling for traditional, SDN and virtual networks," E*ng. Sci. Technol. an Int. J.*, vol. 31, p. 101065, 2022, doi: 10.1016/j.jestch.2021.09.011.

[3] M. P. Karpowicz, "Adaptive tuning of network traffic policing mechanisms for DDoS attack mitigation systems," *Eur. J. Control*, vol. 61, pp. 101–118, 2021, doi: 10.1016/j.ejcon.2021.07.001.

[4] A. Iranmanesh and H. R. Naji, "A protocol for cluster confirmations of SDN controllers against DDoS attacks," *Comput. Electr. Eng*., vol. 93, p. 107265, 2021, doi: 10.1016/j.compeleceng.2021.107265.
[5] A. Sanmorino, R. Gustriansyah, and J. Alie, "DDoS Attacks Detection Method Using Feature Importance and Support Vector Machine," *JUITA*, vol. 10, no. 2, p. 167, Nov. 2022, doi: 10.30595/juita.v10i2.14939.
[6] D. Kumar, R. K. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, "DDoS Detection using Deep Learning," *Procedia Computer Science*, vol. 218, pp. 2420–2429, 2023, doi: 10.1016/j.procs.2023.01.217.
[7] H. C. Altunay and Z. Albayrak, "A hybrid CNN + LSTMbased intrusion detection system for industrial IoT networks," *Eng. Sci. Technol. an Int. J*., vol. 38, p. 101322, 2023, doi: 10.1016/j.jestch.2022.101322.
[8] H. Huang, P. Ye, M. Hu, and J. Wu, "A multi-point collaborative DDoS defense mechanism for IIoT environment," *Digit. Commun. Networks*, vol. 9, no. 2, pp. 590–601, 2023, doi: 10.1016/j.dcan.2022.04.008.
[9] M. Guarascio, G. Manco, and E. Ritacco, "Deep learning," *Encycl. Bioinforma. Comput. Biol. ABC Bioinforma*., vol. 1–3, no. Dl, pp. 634–647, 2018, doi: 10.1016/B978-0-12-809633-8.20352-X.
[10] X. Ren, H. Gu, and W. Wei, "Tree-RNN: Tree structural recurrent neural network for network traffic classification," E*xpert Syst. Appl*., vol. 167, p. 114363, 2021, doi: 10.1016/j.eswa.2020.114363.
[11] M. A. Talukder *et al*., "An efficient deep learning model to categorize brain tumor using reconstruction and fine-tuning," *Expert Syst. Appl*., vol. 230, p. 120534, 2023, doi: 10.1016/j.eswa.2023.120534.
[12] F. Du, J. Zhang, N. Ji, G. Shi, and C. Zhang, "An effective hierarchical extreme learning machine based multimodal fusion framework," *Neurocomputing*, vol. 322, pp. 141–150, 2018, doi: 10.1016/j.neucom.2018.09.005.
[13] X. Liu, C. Wang, J. Bai, and G. Liao, "Fine-tuning Pre-trained Convolutional Neural Networks for Gastric Precancerous Disease Classification on Magnification Narrow-band Imaging Images," *Neurocomputing*, vol. 392, pp. 253–267, 2020, doi: 10.1016/j.neucom.2018.10.100.
[14] F. Ullah, S. Ullah, G. Srivastava, and J. C.-W. Lin, "IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic," *Digit. Commun. Networks*, 2023, doi: 10.1016/j.dcan.2023.03.008.
[15] N. Ahuja, G. Singal, D. Mukhopadhyay, and N. Kumar, "Automated DDOS attack detection in software defined networking," *Journal of Network and Computer Applications*, vol. 187, p. 103108, Aug. 2021, doi: 10.1016/j.jnca.2021.103108.
[16] A. Deshpande, V. V. Estrela, and P. Patavardhan, "The DCT-CNN-ResNet50 architecture to classify brain tumors with super-resolution, convolutional neural network, and the ResNet50," *Neuroscience Informatics*, vol. 1, no. 4, p. 100013, Dec. 2021, doi: 10.1016/j.neuri.2021.100013.
[17] L. M. Tetzlaff and G. Szepannek, "mlr3shiny—State-of-the-art machine learning made easy," *SoftwareX*, vol. 20, p. 101246, Dec. 2022, doi: 10.1016/j.softx.2022.101246.
[18] I. Zoppis, G. Mauri, and R. Dondi, *Kernel methods: Support vector machines*, vol. 1–3. Elsevier Ltd., 2018. doi: 10.1016/B978-0-12-809633-8.20342-7.
[19] G. Stavropoulos, R. van Voorstenbosch, F.-J. van Schooten, and A. Smolinska, *Random Forest and Ensemble Methods*, 2nd ed. Elsevier Inc., 2020, doi: 10.1016/b978-0-12-409547-2.14589-5.
[20] I. S. Polonskaia, I. R. Aliev, and N. O. Nikitin, "Automated Evolutionary Design of CNN Classifiers for Object Recognition on Satellite Images," *Procedia Computer Science*, vol. 193, pp. 210–219, 2021, doi: 10.1016/j.procs.2021.10.021.
[21] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT," *Procedia Computer Science*, vol. 167, pp. 1561–1573, 2020, doi: 10.1016/j.procs.2020.03.367.
[22] B. M. Serinelli, A. Collen, and N. A. Nijdam, "Training Guidance with KDD Cup 1999 and NSL-KDD Data Sets of ANIDINR: Anomaly-Based Network Intrusion Detection System," *Procedia Computer Science*, vol. 175, pp. 560–565, 2020, doi: 10.1016/j.procs.2020.07.080.
[23] K. A. Alnowibet, I. Khan, K. M. Sallam, and A. W. Mohamed, "An efficient algorithm for data parallelism based on stochastic optimization," *Alexandria Eng. J*., vol. 61, no. 12, pp. 12005–12017, 2022, doi: 10.1016/j.aej.2022.05.052.
[24] B. Erkayman, E. Erdem, T. Aydin, and Z. Mahmat, "New Artificial intelligence approaches for brand switching decisions," *Alexandria Eng. J*., vol. 63, pp. 625–643, 2023, doi: 10.1016/j.aej.2022.11.043.
[25] P. Fränti and R. Mariescu-Istodor, "Soft precision and recall," *Pattern Recognit. Lett*., vol. 167, pp. 115–121, 2023, doi: 10.1016/j.patrec.2023.02.005.
[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*., pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
[27] T. Gevers and A. Smeulders, "Foreword," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9909, p. 5, 2016, doi: 10.1007/978-3-319-46493-0.
[28] A. Gupta, P. Pawade, and R. Balakrishnan, "Deep Residual Network and Transfer Learning-based Person Re-Identification," *Intell. Syst. with Appl*., vol. 16, p. 200137, 2022, doi: 10.1016/j.iswa.2022.200137.
[29] B. S. A. Gazioğlu and M. E. Kamaşak, "Effects of objects and image quality on melanoma classification using deep neural networks," *Biomed. Signal Process. Control*, vol. 67, no. July 2020, pp. 1–9, 2021, doi: 10.1016/j.bspc.2021.102530.
[30] C. Saisree and D. K. U., "Pothole Detection Using Deep Learning Classification Method," *Procedia Comput. Sci*., vol. 218, pp. 2143–2152, 2023, doi: 10.1016/j.procs.2023.01.190.
[31] S. Belattar, O. Abdoun, and E. K. Haimoudi, "Performance analysis of the application of convolutional neural networks architectures in the agricultural diagnosis," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 1, pp. 156–162, Jul. 2022, doi: 10.11591/ijeecs.v27.i1.pp156-162.
[32] M. Thu, N. Suvonvorn, and N. Kittiphattanabawon, "Pedestrian classification on transfer learning based deep convolutional neural network for partial occlusion handling," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 3, pp. 2812–2826, Jun. 2023, doi: 10.11591/ijece.v13i3.pp2812-2826.
[33] T. El Moudden, R. Dahmani, M. Amnai, and A. A. Fora, "Slum image detection and localization using transfer learning: a case study in Northern Morocco," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 3, pp. 3299–3310, 2023, doi: 10.11591/ijece.v13i3.pp3299-3310.
[34] F. J. Abdullayeva, "Distributed denial of service attack detection in E-government cloud via data clustering," *Array*, vol. 15, , p. 100229, 2022, doi: 10.1016/j.array.2022.100229.
[35] M. Wang, Y. Lu, and J. Qin, "A dynamic MLP-based DDoS attack detection method using feature selection and feedback," *Comput. Secur.*, vol. 88, 2020, doi: 10.1016/j.cose.2019.101645.
[36] A. Sanmorino, R. Gustriansyah, Terttiaavini, and Isabella, "The toolkit of success rate calculation of broiler harvest," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 15, no. 4, pp. 1947–1954, 2017, doi: 10.12928/TELKOMNIKA.v15i4.6744.
[37] A. Gaurav, B. B. Gupta, and P. K. Panigrahi, "A novel approach for DDoS attacks detection in COVID-19 scenario for small entrepreneurs," *Technological Forecasting and Social Change*, vol. 177, p. 121554, Apr. 2022, doi:

10.1016/j.techfore.2022.121554.
[38] O. Thorat, N. Parekh, and R. Mangrulkar, "TaxoDaCML: Taxonomy based Divide and Conquer using machine learning approach for DDoS attack classification," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100048, 2021, doi: 10.1016/j.jjimei.2021.100048.
[39] R. Poojary, R. Raina, and A. K. Mondal, "Effect of data-augmentation on fine-tuned cnn model performance," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 1, pp. 84–92, 2021, doi: 10.11591/ijai.v10.i1.pp84-92.
[40] C. X. Ge, M. A. As'ari, and N. A. J. Sufri, "Multiple face mask wearer detection based on YOLOv3 approach," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 1, pp. 384–393, 2023, doi: 10.11591/ijai.v12.i1.pp384-393.

BIOGRAPHIES OF AUTHORS

Ahmad Sanmorino is a senior lecturer at postgraduate computer science, Faculty of Computer and Science, Universitas Indo Global Mandiri. He received a doctorate degree in Informatic Engineering from Universitas Sriwijaya. His research interests lie in artificial intelligence area including applied machine learning, cyber security, and information security. He can be contacted at email: sanmorino@uigm.ac.id.

Hendra Di Kesuma is a lecturer at the Faculty of Computer and Science, Universitas Indo Global Mandiri. He received a master's degree in Computer Science from Universitas Gadjah Mada. His research interests lie in the fields of information systems, information security, and web technology. He can be contacted at email: hendra.dikesuma@uigm.ac.id.