Computer Science and Information Technologies
Vol. 6, No. 2, July 2025, pp. 202~213
ISSN: 2722-3221, DOI: 10.11591/csit.v6i2.p202-213

Attack detection in internet of things networks with deep learning using deep transfer learning method

Riki Abdillah Hasanuddin[1], Muhammad Subali[2]
[1]Department Electrical Engineering, Faculty of Technology and Engineering, Gunadarma University, Jakarta, Indonesia
[2]Faculty of Informatics Engineering, Cendekia Abditama University, Tangerang, Indonesia

Article Info

Article history:

Received Dec 15, 2024
Revised Apr 22, 2025
Accepted Apr 29, 2025

Keywords:

1D-CNN
Attack detection
Deep learning
IoT network
Transfer learning

ABSTRACT

Cybersecurity becomes a crucial part within the information management framework of internet of things (IoT) device networks. The large-scale distribution of IoT networks and the complexity of communication protocols used are contributing factors to the widespread vulnerabilities of IoT devices. The implementation of transfer learning models in deep learning can achieve optimal performance faster than traditional machine learning models, as they leverage knowledge from previous models that already understand these features. Base model was built using the 1-dimension convolutional neural network (1D-CNN) method, using training and test data from the source domain dataset. Model 1 was constructed using the same method as base model. The test and training data used for model 1 were from the target domain dataset. This model successfully detected known attacks at a rate of 99.352%, but did not perform well in detecting unknown attacks, with an accuracy of 84.645%. Model 2 is an enhancement of model 1, incorporating transfer learning from the base model. Its results significantly improved compared to model 1 testing. Model 2 has an accuracy and precision rate of 98.86% and 99.17 %, respectively, allowing it to detect previously unknown attacks. Even with a slight decrease in normal detection, most attacks can still be detected.

Corresponding Author:

Muhammad Subali
Department of Computer Science and Information Technology, Cendekia Abditama University
Islamic Raya St., Kelapa Dua, Tangerang, Banten, Indonesia
Email: subali@uca.ac.id

1. INTRODUCTION

The development of information technology, particularly internet of things (IoT) devices, is progressing rapidly and is directly proportional to the increasing number of IoT device users, both individuals and industries. The role of IoT devices has influenced all areas directly related to technology and business, enhancing benefits for both individuals and organizations [1]. IoT devices enhance user experiences by providing immediate data access yet introduce a multitude of cybersecurity vulnerabilities across different operational layers. Cyber-attacks on these devices can be categorized as goal-oriented attacks (targeting objectives like unauthorized access or data exfiltration), performance-oriented attacks (such as DoS/DDoS efforts that degrade system availability), and layer-oriented attacks (exploiting weaknesses at the edge, access/middleware, or application layers). Addressing these threats requires tailored security strategies—from fortified cloud services and resilient 5G network designs to protections that account for IoT devices' heterogeneous nature and limited computing resources.[2]. The application of transfer learning methods in deep learning has been widely used to train models and is effective in identifying attacks. Transfer learning models can achieve optimal performance more quickly than traditional machine learning

models because these models leverage knowledge (features and weights) from previous models that already understand these features, making it faster than training a neural network from scratch. Additionally, transfer learning is more computationally efficient and helps achieve better results using a small dataset [3].

Clustering-enhanced transfer learning approach (CeHTL), an enhanced approach that automatically determines the relationship between new and known attacks [4]. An improved convolutional neural network (ICNN) characterizes and preprocesses network traffic data, extracts advanced features, and optimizes parameters with stochastic gradient descent [5]. Deep transfer learning method that uses two autoencoders to align feature representations and effectively detect IoT attacks, outperforming other approaches on nine IoT datasets [6]. Intrusion detection model that uses convolutional neural networks in 1D, 2D, and 3D, along with transfer learning to handle binary and multiclass classification [7]. Efficient network-based intrusion detection system for IoT networks that combines a deep neural network (DNN) with mutual information (MI)-based feature selection to detect anomalies and zero-day cyberattacks.[8]. Transfer learning approach to updating intrusion detection systems (IDS) that have become outdated due to their heavy reliance on initial training datasets and their inability to detect changes in attacks [9]. Deep transfer learning framework using weight transferring and neural network fine-tuning for end-to-end learning, addressing concept drift and reducing human intervention [10]. The experiments showed that the individual DNN, RNN, or CNN approaches are better than the combined models (CNN+RNN and CNN+LSTM) [11]. A transfer learning framework using an optimal source domain dataset improves network-level intrusion detection [12]. Federated transfer learning (FTL) framework for IIoT network intrusion detection uses a neural network that distributes IoT data processing between client and server devices [13]. Deep transfer learning approach for rolling bearing fault diagnosis using a 1D-CNN extracts features from vibration signals and uses CORrelation ALignment (CORAL) to minimize the marginal distribution discrepancy between source and target domains [14]. The suitability of deep learning for anomaly-based IDS by developing models using various deep neural network architectures including CNNs, AEs, and RNNs [15]. Transfer learning–based IDS for cloud-based IoT environments, addressing the increased security risks inherent in centralized data processing [16]. Unified indoor–outdoor localization solution for IoT devices in smart cities using an encoder-based transfer learning scheme, the proposed approach builds a single deep learning model that adapts across both indoor and outdoor settings, reducing complexity and costs [17]. Sequential intrusion detection system that leverages deep learning techniques specifically, Text-CNN and GRU to extract features from the network layer and the application layer, treating sequential data like a language model.[18]. In relation to the studies, we proposed model 1D-CNN architecture in Ubuntu environment. The main purpose of this research is to produce a model using deep transfer learning methods that is trained with a source domain dataset, where the model can detect both known and unknown attacks in the target domain dataset with small dataset. In other research, two-dimensional convolutional neural network (CNN2D) architecture is used to build model in Windows environment [19].

2. METHOD

2.1. Background theory

Transfer learning is an important tool in machine learning to address the fundamental problem of insufficient training data. It attempts to transfer knowledge from the source domain to the target domain by relaxing the assumption that training and testing data must be integrated, identically distributed (IID). This will lead to significant positive effects for many domains that are difficult to improve due to a lack of training data. The definition of transfer learning can be described as follows: a learning task is assigned to Tt based on Dt and can receive assistance from Ds for the learning task Ts. Transfer learning aims to enhance the performance of the predictive function fT(.) for the learning task Tt by identifying and transferring latent knowledge from Ds and Ts where Ds=Dt and/or Ts=Tt. In many cases, the size of Ds is larger than Dt, Ns>>Nt [20].

The concept of IoT was created by a member of the radio frequency identification (RFID) development community in 1999. Generally, IoT is defined as a network of physical objects. The internet is not only a network of computers but has evolved into a network of devices of all kinds and sizes, including vehicles, smartphones, household appliances, medical equipment, industrial systems, humans, animals, buildings, all connected and communicating and sharing information based on established protocols to achieve intelligent reorganization, placement, tracking, security, and control, as well as personal online monitoring, process control, and administration [21]. The number of IoT devices is rapidly increasing, and the lack of security in these devices has made them targets for criminal activities. Figure 1 presents the variations of cyber security attacks that occur at the IoT layers such as the perception, support, network, and application layers [22].

CNN is a highly powerful class of deep learning that is widely applied in various tasks, including object detection, speech recognition, computer vision, image classification, bioinformatics, and time series

prediction tasks. CNNs are feedforward neural networks that utilize convolutional structures to extract features from data. Unlike traditional methods, CNNs automatically learn and recognize features from data without requiring manual feature extraction by humans. The design of CNNs is inspired by visual perception. The main components of CNNs include convolutional layers, pooling layers, and fully connected layers. The convolution layer is a crucial component of CNNs. Through several convolutional layers, the convolution operation extracts different features from the input. Pooling layer: typically following the convolution layer, the pooling layer reduces the number of connections in the network by performing dimensionality reduction on the input data. Its primary objective is to decrease the computational load and address the issue of overfitting. The pooling operation produces an output feature map that is more resilient to distortions and errors in the neurons. Fully connected: the fully connected layer is typically located at the end of the CNN architecture. In this layer, each neuron is connected to all neurons in the previous layer, following the principles of conventional multi-layer perceptron neural networks. The fully connected layer receives input from the last pooling or convolution layer, which is a vector created by flattening the feature maps. The fully connected layer serves as the classification component in CNNs, enabling the network to make predictions [23].

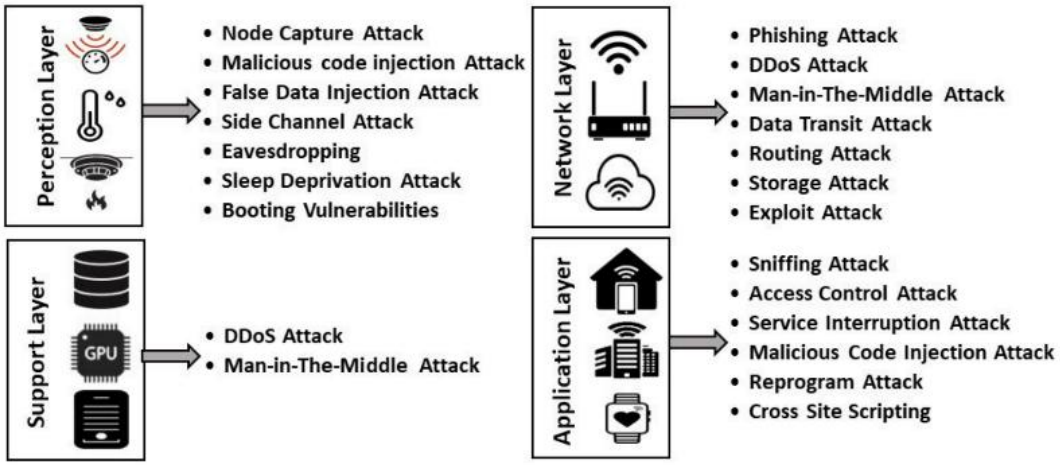

Figure 1. Variations in cybersecurity attacks at the IoT layer [22]

2.2. Methodology

In this research, the methodology employed is divided into 5 stages, which are explained as follows:

– Phase 1: data preprocessing IoT BoT dataset
– Phase 2: data preprocessing UNSW-NB15 dataset
– Phase 3: build and train – test model 1 with IoT BoT dataset for base model and UNSW-NB15 for testing model 1 without transfer learning
– Phase 4: build and train – test model 2 which model 1 updated with transfer learning form base model
– Phase 5: evaluation model 2

The research methodology diagram can be seen in Figure 2.

2.3. Data preparation, treatment, and preprocessing

In this research, development of the model focuses on building a model capable of detecting and classifying various types of IoT attacks. In this regard, two datasets are utilized for the source and transfer domains containing streams of normal IoT traffic and cyber-attacks. BoT-IoT dataset is used for the source domain because it contains a substantial amount of IoT network activity data, whereas UNSW-NB15 dataset is used for the target domain due to its rarity and disproportionate representation of IoT network traffic comprised of cyber-attacks. To evaluate attack detection, four different datasets will be created as described.

Bot-IoT dataset as source domain dataset has a volume of 73 million sample [24]. For this research, 5% of the dataset will be used, totaling approximately 3.6 million samples [25]. The 5% dataset includes normal traffic samples and 4 categories of attacks, namely denial of service (DoS), distributed denial of service (DDoS), reconnaissance, and information theft which shown in Table 1.

UNSW-NB15 as target domain dataset [26], created using the IXIA PerfectStorm tool in the Cyber Range Lab at UNSW Canberra to generate a combination of modern normal activities and synthetic contemporary attack behaviors. The Argus tool and Bro-IDS were utilized, and 12 algorithms were developed to

produce 49 features labeled as normal and attack. The attack categories were classified into 9 groups based on the nature of the attacks [27]. The total number of records is 2,540,047 which shown in Table 2.

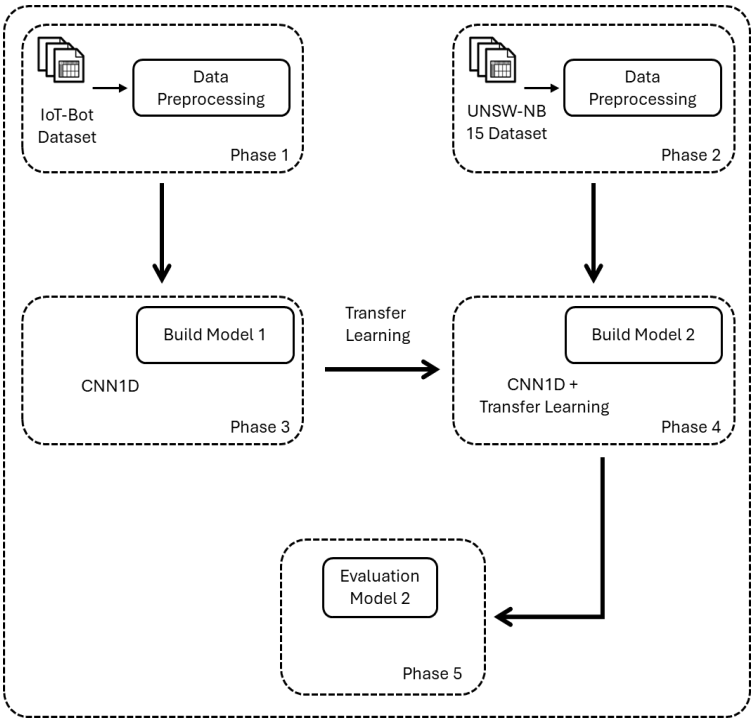

Figure 2. Research methodology diagram

Table 1. Distribution of Bot-IoT 5% dataset

Name	Subcategory	Description	Record	%
Normal	Normal	Natural transaction	477	0.00
DDos	TCP UDP HTTP	Attack where multiple compromised computer systems attack a target, causing a DOS	1,926,624	52.51
Dos	TCP UDP HTTP	A malicious attack to cripple the services offered by a site, server or network overloading the target of its associated by flooding the site with many requests	1,650,260	44.97
Reconnaissance	OS Fingerprinting Service Scanning	All the different strikes simulating attacks gathering information	91,882	2.50
Information theft	Keylogging Data exfiltration	Stealing of personal user information	79	0.00

Table 2. Distribution of UNSW-NB15 dataset

Name	Description	Record	%
Normal	Natural transaction data	2,218,764	87.35
Generic	Attack against blockciphers with a given block and key size (not considering its structure)	215,481	8.48
Exploits	Attack that exploits vulnerabilities, taking advantage of security problems (of an operating system or a piece of software) known by the attackers.	44,525	1.75
Fuzzers	Attack that suspends a program or network, feeding it with randomly generated data.	24,246	0.95
Dos	A malicious attack that makes a server or network resource unavailable, overloading the target of the associated infrastructure with a flood of Internet traffic.	16,353	0.64
Reconnaissance	Comprises different attacks that gather information.	13,987	0.55
Analysis	Different attacks on penetrations (HTML files, spam, and port scan)	2,677	0.11
Backdoor	An attack that bypasses a system security mechanism to access a computer or its data.	2,329	0.09
Shellcode	Attack that exploits software vulnerabilities using small pieces of code as payloads.	1,511	0.06
Worms	Attack where the attacker replicates itself to spread to other computers.	174	0.01

As indicated by the distribution of UNSW-NB15, which exhibits a ratio of 87% normal traffic and 13% attack traffic, the system functions in a balanced manner between these two traffic categories. This was subsequently balanced to approximate a realistic scenario, resulting in 50% of normal traffic and 50% of attacks. To ascertain the efficacy of the transfer learning-based model in detecting both known and unknown attacks, three distinct datasets were generated in the target domain and are presented in Table 3.

a) UNSW-NB15_base4: dataset containing normal traffic and three type of known attacks (generic, DoS, and reconnaissance) used for training and divided into two subdataset:
   – UNSW-NB15_base4_train : 75% of UNSW-NB15_base4 to train model 1.
   – UNSW-NB15_base4_test : 25% of UNSW-NB15_base4 to evaluate the effectiveness in detection of known attacks.
b) UNSW-NB15_first-test: dataset to evaluate effectiveness in detection of unknown attacks (exploits, fuzzers, analisis, backdoor, shellcode, and worms).
c) UNSW-NB15_full-test: dataset to evaluate effectiveness in detection of attack type known attacks (generic, DoS, and reconnaissance) and unknown attacks (exploits, fuzzers, analisis, backdoor, shellcode, and worms).

It is important to note that phases 1 and 2 of the frameworks involve data processing. In the transfer learning-based model under design, the output from the convolutional base is utilized as input for the classifier. Subsequently, the datasets for the source and target domains must be trained with the same input formats and features. For this study, new versions were generated for both datasets with their 15 common features, which are shown in Table 4.

In this phase, the columns with string formats are converted to numeric formats using the one hot encoding (OHE) method. The columns formatted in hexadecimal (HEX) are converted to decimal (DEC) format. A logarithmic procedure is applied to the columns (ie., dur, sbytes, dbytes, and spkts) which have concentrated values at 0. Standard normalization of the dataset is performed to prevent overfitting and potential bias in the results. Finally, the data is transformed into an image format into a 1D format (i.e., a vector of length 24 yielding dimensions of [1], [24]).

Table 3. Distribution of UNSW-NB15 subdataset

	UNSW-NB15_base4_train		UNSW-NB15_base4_test		UNSW-NB15_first-test		UNSW-NB15_full-test	
Name	Record	%	Record	%	Record	%	Record	%
Normal	183,969	51.67	61,465	50.01	75,462	50.00	321,283	50.00
Generic	160,198	44.99	53,938	43.88			215,481	33.53
Exploits					44,525	29.50	44,525	6.93
Fuzzers					24,246	16.07	24,246	3.77
Dos	8,971	2.52	4,073	3.31			16,353	2.54
Reconnaissance	2,909	0.82	3,434	2.79			13,987	2.18
Analysis					2,677	1.77	2,677	0.42
Backdoor					2,329	1.54	2,329	0.36
Shellcode					1,511	1.00	1,511	0.24
Worms					174	0.12	174	0.03

Table 4. 15 common features in both dataset

No	Bot-IoT	UNSWNB-15	Type	Description
1	Proto	Proto	Nominal	Textual representation of transaction protocols presents in network flow
2	Saddr	Srcip	Nominal	Source IP address.
3	Sport	Sport	Integer	Source port number.
4	Daddr	Dstip	Nominal	Destination IP address.
5	Dport	Dsport	Integer	Destination Port number.
6	Spkts	Spkts	Float	Source-to-destination packet count
7	Dpkts	Dpkts	Float	Destination-to-source packet count.
8	Sbytes	Sbytes	Float	Source-to-destination byte count
9	Dbytes	Dbytes	Float	Destination-to-source byte count.
10	State	State	Nominal	Transaction state.
11	Stime	Stime	Timestamp	Record start time.
12	Ltime	Ltime	Timestamp	Record end time.
13	Dur	Dur	Float	Record total duration.
14	Attack	Label	Binary	Class label: 0 for normal traffic, 1 for attack.
15	Category	attack_cat	Nominal	Cyberattack family.

2.4. Transfer learning and model design

In phase 3, model 1 is constructed as a base model using the BoT-IoT dataset, which has been randomly partitioned into 75% training data and 25% testing data. In phase 4, model 1 experienced an update

that integrated the knowledge acquired in the source domain into the target domain, resulting in the formation of model 2. In model 2, the convolutional base of the original base model is fixed, and the classifier is fed with its outputs. Model 2 is trained using the UNSW-NB15_Base4_train dataset, which represents 75% of the UNSW-NB15_Base4 dataset. In the final phase of the evaluation, model 2's capabilities to detect both known and unknown attacks will be assessed by using the UNSW-NB15_first_test and UNSW-NB15_full_test datasets. This approach encompasses both known and unknown attack scenarios, thereby ensuring a comprehensive assessment of Model 2's capabilities.

This study designs and develops a 1D-CNN for attack detection in IoT networks. Model 1 serves as the base model, and model 2 adds knowledge from it. The proposed models are outlined in Figure 3. The first model (hereafter referred to as "model 1") consists of an input layer, two blocks of convolution layers, a flatten layer, a fully connected layer, and an output layer. Each block comprises a convolutional layer, a normalization layer, a pooling layer, and a dropout layer. The convolution layer is a critical component of deep learning algorithms, responsible for extracting features from raw data. It accomplishes this by learning data attributes from small sub-samples of the input data, ensuring the preservation of the underlying associations between data points. The objective of layer normalization is to normalize all inputs to a specific neural network layer. The layer normalization layer is responsible for standardizing the output of the convolution layer for the max pooling layer. The max pooling layer is responsible for determining the total number of features present in each patch. It identifies the features that demonstrate the greatest dominance within specific localized regions. This process enables the network to prioritize the most significant elements while effectively reducing redundancy. A spatial dropout layer is employed to eliminate the entire feature map rather than individual units. The flatten layer is fully connected to a dense layer. The final layer of the model is designated as the output layer for binary classification.

Model 2 represents an updated version of model 1. It comprises an input layer, a frozen layer with transfer learning for the base model, a flatten layer, three blocks of fully connected layers with a dropout layer, and an output layer. The input layer contains an equal number of features. During the training of the binary classification model, the convolution layers, normalization layers, pooling layers, dropout layers, and flatten layers were frozen. During the training phase, learning was permitted exclusively in the dense and output layers. As demonstrated in Figure 3, a general perspective from both models is presented.

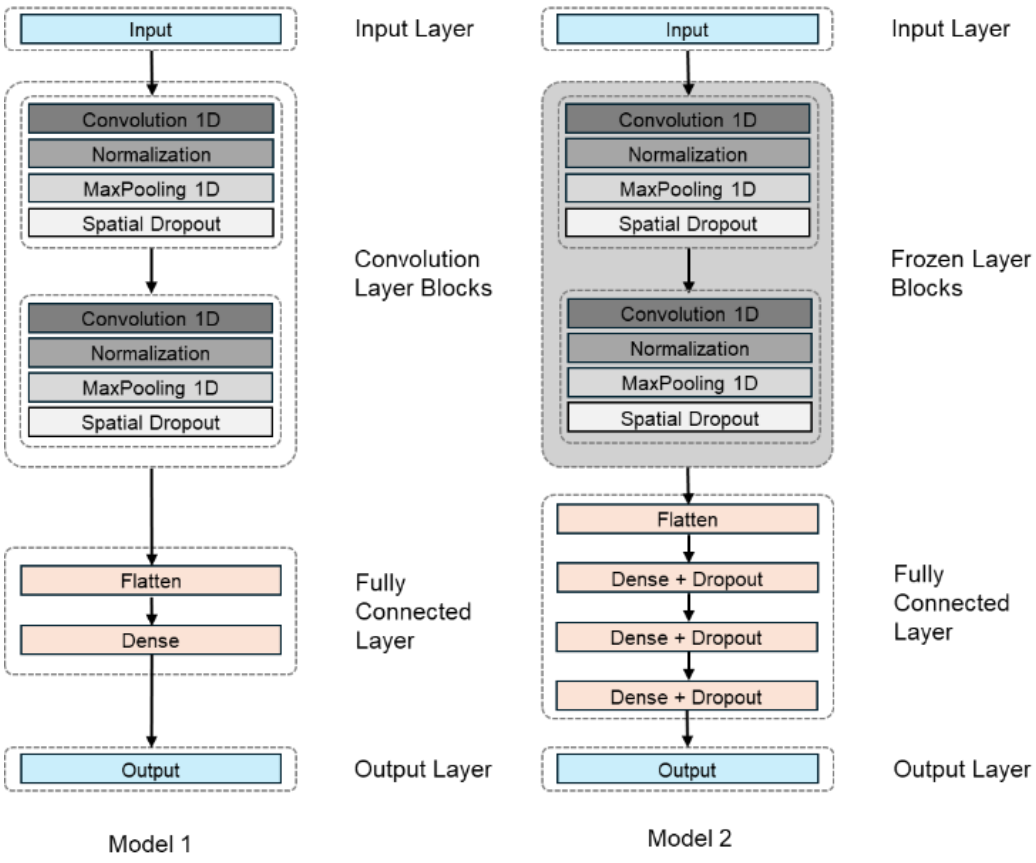

Figure 3. General view of model 1 and model 2

In model 1, the input for 1D-CNN is two-dimensional. Initially, an input vector [1], [24] is generated to accommodate the 15 common features in phases 1 and 2. Subsequent to the incorporation of the input layer, the model was augmented with four convolution layer blocks. The convolution operation in the first layer is configured with a 64-filter configuration, a 24-filter kernel size, a rectified linear unit (ReLU) activation function, and the same padding parameter. The layer normalization adjusts the preceding layer activation separately for each sample in a given batch. The maximum pooling layer provides a solution that ensures the preservation of the most salient features, thereby enhancing the efficiency of the training process and improving the performance of the model. A spatial dropout layer is employed to regularize the training data model and mitigate overfitting, with a drop value of 0.05. It is noteworthy that each of the four convolution layers employs identical parameters. The classification component comprises fully connected, flattened, and dense layers. The flatten layer is applied to the model, thereby transforming the tensor into a shape that is equivalent to the tensor elements. The flatten layer is connected to a fully connected dense layer, and the dense layer is connected to the output layer. The dense layer contains 512 neurons, and a single neuron is responsible for implementing the sigmoid activation function in the output layer. The model has been trained over the course of 15 epochs, with a batch size of 256 and an Adam optimizer with a learning rate of 2×10^{-5} This training process has been implemented to minimize the error function and the binary cross-entropy loss function.

In model 2, the convolution layer from the model 1 is to be frozen in order to prevent weight changes during training. This layer is to be followed by three blocks of fully connected layers. The first layer consists of 512 neurons with a drop value of 0.4, the second layer consists of 256 neurons with a drop value of 0.3, and the last layer consists of 128 neurons with a drop value of 0.2. The output layer is comprised of a single neuron that exhibits sigmoid activation. The model has been trained over 50 epochs, with a batch size of 4096, an Adam optimizer with a learning rate of 6×10^{-5}, and a binary cross-entropy loss function. The training parameters for both models are summarized in Table 5.

2.5. Evaluation metrics

In phase 5, the two models are validated using accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC) score. Accuracy is defined as the proportion of samples that are correctly identified, expressed as a percentage of the total number of samples. Precision is quantified by the proportion of instances that are accurately classified to the total true positive (TP) and false positive (FP) cases. The calculation of recall is determined by the division of the total number of TP measurements by the total TP and false negative (FN) measurements. The F1-score is determined by the calculation of the weighted average of precision and recall. Furthermore, the CNN model's validation is supported by the false positive rate (FPR), defined as the number of normal samples that are classified as positive, and the false negative rate (FNR), denoting the number of abnormal samples that are identified as negative. The Matthews correlation coefficient (MCC) is a metric that considers true and false positives and negatives, thereby providing a balanced measure of classification performance. The range of possible values is from -1 to 1. In this context, 1 indicates perfect prediction, 0 indicates no better than random chance, and -1 indicates total disagreement between prediction and true prediction [28].

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (1)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (2)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (3)$$

$$F1\ score = 2 \times \frac{(Precision*Recall)}{(Precision+Recall)} \quad (4)$$

$$MCC = \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}} \quad (5)$$

2.6. System setup

The transfer learning experimental environment was constructed on an Asus X441UB laptop, which is supported by the following hardware specifications: an Intel® Core™ i3-6006U CPU @ 2.0 GHz processor and 8 GB of RAM DDR4. This laptop model is supported by an NVIDIA® GeForce® MX110 2 GB GDDR3 graphics card with a GPU driver, cuda 12.2.2, CuDNN 8.9.7.29, and TensorRT 10.2.0. The

operating system utilized in this study is Ubuntu 22.04 OS, which is equipped with Python 3.11.0, Tensorflow 2.15.1, Scikit-Learn, and Keras packages.

Table 5. The parameter of model 1 and model 2

Layer	Name	Model 1		Model 2	
		Number	Description	Number	Description
Input	Input layer	1	24 fitur	1	24 fitur
Hidden layer	Conv1D	2	Filter =64, Padding = same, Kernel_size =24, Activation = ReLu		
	Normalization	2			
	MaxPool1D	2	Pool_size =2, Padding = same		
	Spatial dropout	2	Rate =0.05		
Classification	Dense	1	Neuron =512, Activation = Relu	1	Neuron =512, Activation = Relu, Dropout =0.4
				1	Neuron =256, Activation = Relu, Dropout =0.4
				1	Neuron =128, Activation = Relu, Dropout =0.2
Output	Output layer	1	Neuron=1, Activation = Sigmoid	1	Neuron =1, Activation = Sigmoid
Hyperparameters	Early stopping (monitor = val_loss, patience =5) Optimizers = Adam Loss function =BinaryCrossentrophy Learning rate =2e-5 Batch size =256 Epoch =15			Early stopping (monitor = val_loss, patience =5), optimizers = Adam Loss function = BinaryCrossentrophy Learning rate =6e-5 Batch size =4096 Epoch =50	

3. RESULTS AND DISCUSSION

3.1. Evaluation model 1 as base model

Model 1 as base model is trained using the 5% BoT-IoT has an optimal attack detection rate with accuracy, precision, recall, and F1-score achieve 100% and MCC of 0.9502 meaning that almost all traffic can be detected accurately. However, some normal traffic is identified by the model as attack traffic, and some attack traffic, particularly reconnaissance type traffic, is detected as normal traffic. The complete results of traffic detection from this model are shown in Tables 6 and 7.

Table 6. Performance of base model

Name	Detected	Not detected	% Detected
Normal	104	2	98.11
DDos	385,282	0	100.00
Dos	330,097	0	100.00
Reconnaissance	16,438	9	99.95
Information theft	16	0	100.00

Table 7. Metrics of base model

Metrics	Value
Accuracy	100.00%
Precision	100.00%
Recall	100.00%
F1-Score	100.00%
MCC	0.9502

3.2. Evaluation model 1 without transfer learning

In this stage, model 1 is trained using the UNSW-NB15-Base4_train dataset and tested using the UNSW-NB15-first_test dataset. The dataset contains types of attacks that have not been trained (unknown attacks), allowing for the assessment of model 1's ability to detect new types of attacks. Model 1 is capable of detecting normal traffic well, but for unknown attack, especially fuzzers and analysis, it has a low detection rate of 48.17% and 41.64% respectively. Furthermore, model 1 will be tested with UNSW-NB15_full_test to determine the extent to which detecting known attacks and unknown attacks. The capability of model 1 for attack detection is quite good. However, there are still types of attacks that remain inadequately detectable, such as backdoor and shellcode, it has a low detection rate of 82.07% and 84.65%. The complete results of traffic detection from model 1 are shown in Table 8.

3.3. Evaluation model 2 (model 1 with transfer learning)

At this stage, model 2 is constructed where model 1 is enhanced with transfer learning, allowing for understanding that the use of transfer learning can assist model 2 in detecting both known and unknown attacks more effectively compared to model 1. In first test using UNSW-NB15_first_test dataset, model 1 has ability to detect unknown attacks and has improved because of transfer learning compared to without it. Only the fuzzers and worms have a detection rate of 96%, while others have rates above 97%. The second test will

use the UNSW-NB15_full_test dataset, which includes both known and unknown attacks. For unknown attack types, analysis has the highest detection rate at 99.84%, while backdoors have the lowest detection rate at 89.36%. For known attack detections, reconnaissance has the highest detection rate at 99.94%, and exploits have the lowest detection rate at 94.53%. The complete results of the detection rate from model 2 are shown in Table 9.

Table 8. Detection rate in model 1 without transfer learning

Name	UNSW-NB15-first_test			UNSW-NB15_full_test		
	Detected	Not detected	% Detected	Detected	Not detected	% Detected
Normal	74,518	794	98.95	316,086	4,507	98.59
Generic				213,545	173	99.92
Exploits	18,556	9,761	65.53	25,741	2,576	90.90
Fuzzers	10,365	11,152	48.17	20,643	874	95.94
Reconnaissance				11,835	19	99.84
DoS				3,650	213	94.49
Shellcode	1,297	214	85.84	1,279	232	84.65
Analysis	259	363	41.64	607	15	97.59
Backdoor	244	113	68.35	293	64	82.07
Worms	67	67	50.00	169	5	97.13

Table 9. Detection rate in model 2 (model 1 with transfer learning)

Name	UNSW-NB15-first_test			UNSW-NB15_full_test		
	Detected	Not detected	% Detected	Detected	Not detected	% Detected
Normal	72,903	2,409	96.80	316,039	4,554	98.58
Generic				213,692	26	99.99
Exploits	27,588	729	97.43	26,768	1,549	94.53
Fuzzers	20,725	792	96.32	21,017	500	97.68
Reconnaissance				11,847	7	99.94
DoS				3,779	84	97.83
Shellcode	1471	40	97.35	1,359	152	89.94
Analysis	621	1	99.84	621	1	99.84
Backdoor	350	7	98.04	319	38	89.36
Worms	168	6	96.55	169	5	97.13

3.4. Validation of the model

First, the detection rate from both tests compared and indicated a significant improvement. Overall, in first test, UNSW-NB15_first_test dataset used, each type of unknown attack showed improvement, particularly for analysis, which increased by 58.20% from 41.64% to 99.84%, and fuzzers, which increased by 48.15% from 48.17% to 96.32%, while the detection of normal traffic has decreased from 98.95 to 96.80%. The detection of normal traffic is slightly lower because the training of model 1 was conducted with the BoT-IoT dataset, which has a very small percentage of normal traffic compared to attack traffic. In second test used UNSW-NB15_full_test to assess its capability in detecting both known and unknown attacks. Overall, each type of unknown attack has also shown an increase, with unknown attack types such as backdoors and exploits exhibiting a significant improvement of 7.72% and 5.29% while for known attacks, DoS increased by 3.34% from 94.49% to 97.83%, respectively. However, the detection rate of normal traffic still slightly decreased by -0.01%.

The complete results of the comparison and improvement detection rates from both models are shown in Table 10. The following conclusions can be drawn from this data. It has been demonstrated that Model 2, when employing the transfer learning method, exhibits an enhanced detection rate for both known and unknown attacks. The known attack detection rate ranges from 97.83% to 99.99%, with the unknown attack detection rate ranging from 89.36% to 99.84%. A comparison of models 1 and 2 reveals that the latter, utilizing transfer learning, enhances the detection rate for unknown attacks by up to 58.20%, and for known attacks by up to 7.29%. Therefore, model 2 with transfer learning demonstrates a significant enhancement in the detection of both known and unknown attacks, despite the reduced representation in the dataset, when compared with the model.

Comparison was made between the evaluation metrics from both tests. Model 1 was initially evaluated for its capacity to detect unknowns using the UNSW-NB15_first_test dataset. The model's overall accuracy was found to be 82.42%, with a precision of 58.69%, a recall of 97.49%, an F1-score of 63.64%, and an MCC of 0.6573. Model 1 was also evaluated for the detection of both unknown and known attacks using the UNSW-NB15-full_test dataset. In this particular instance, the accuracy rate was recorded at

98.55%, the precision rate at 98.50%, the recall rate at 98.34%, the F1-score at 87.29%, and the MCC at 0.9709. A comparative analysis of the overall metrics for both solutions reveals that model 2 exhibits superior performance in all metrics when compared to model 1. The accuracy, precision, and F1-score metrics have been enhanced by 14.49%, 38.31%, and 22.95%, respectively, in the detection of unknown attacks. Furthermore, these metrics have been improved by 0.31%, 0.67%, and 0.34%, respectively, when considering both known and unknown attacks.

Evaluation metrics from both tests are compared. Model 1 was evaluated first on the detection of unknown with the UNSW-NB15_first_test dataset. It achieves an overall accuracy of 82.42%, a precision of 58.69%, a recall of 97.49%, an F1-score of 63.64%, and an MCC of 0.6573. Model 1 was also evaluated for the detection of both unknown and known attacks using the UNSW-NB15-full_test dataset. In this case, the accuracy is 98.55%, precision is 98.50 %, recall is 98.34%, F1-score is 87.29%, and MCC is 0.9709. If the overall metrics for both solutions are compared, we can conclude that model 2 outperforms model 1 for all metrics. Accuracy, precision and F1-score are improved by 14.49%, 38.31%, and 22.95% in the detection of unknown, and 0.31%, 0.67%, and 0.34% respectively when considering both known and unknown attacks. The primary factor contributing to this enhancement is the faster convergence of the optimization algorithm to optimal weights in model 2 when the weights are initialized in a similar domain. This contrasts with model 1, which commences from the beginning and utilizes random weights. Another metric that has shown notable improvement is MCC. Model 2 demonstrates higher MCC values due to two primary factors: the occurrence of unknown attacks and the imbalanced nature of the dataset. It can be concluded from these factors that model 2 makes superior predictions in terms of attack detection. This is a critical consideration in adapting the model to different IoT device traffic patterns. As demonstrated in Table 11, this study provides a comprehensive evaluation of the performance metrics from models 1 and 2.

Table 10. Comparison and improvement detection rate in model 1 and model 2

Name	UNSW-NB15-first_test			UNSW-NB15-full_test		
	Model 1 (%)	Model 2 (%)	Improve (%)	Model 1 (%)	Model 2 (%)	Improve (%)
Normal	98.95	96.80	-2.15	98.59	98.58	-0.01
Generic				99.92	99.99	0.07
Exploits	65.53	97.43	31.90	90.90	94.53	3.63
Fuzzers	48.17	96.32	48.15	95.94	97.68	1.74
Reconnaissance				99.84	99.94	0.1
DoS				94.49	97.83	3.34
Shellcode	85.84	97.35	11.51	84.65	89.94	5.29
Analysis	41.64	99.84	58.20	97.59	99.84	2.25
Backdoor	68.35	98.04	29.69	82.07	89.36	7.29
Worms	50.00	96.55	46.55	97.13	97.13	0

Table 11. Comparison of evaluation metrics from model 1 and model 2

Metric	UNSW-NB15-first_test			UNSW-NB15-full_test		
	Model 1	Model 2	Improvement	Model 1	Model 2 (%)	Improvement
Accuracy	82.42%	96.88%	14.49%	98.55%	98.86%	0.31%
Precision	58.69%	97.00%	38.31%	98.50%	99.17%	0.67%
Recall	97.49%	95.48%	-2.01%	98.34%	98.40%	0.06%
F1-Score	73.27%	96.22%	22.95%	98.44%	98.72%	0.34%
MCC	0.6573	0.9358	0.2785	0.9709	0.9770	0.0061

4. CONCLUSION

In this research, we investigate the detection capabilities of transfer learning-based methods for both unknown and known attacks in IoT networks with scarce and unbalanced datasets. To this end, we develop an efficient attack detection framework that combines knowledge transfer and model refinement, achieving high detection accuracy for both known and unknown attacks. The BoT-IoT dataset is utilized to learn knowledge (source domain) and is applied to the UNSW-NB15 dataset (target domain). To evaluate the performance and feasibility of the proposed transfer learning-based model, we generate UNSW-NB15_first-test dataset to evaluate effectiveness in detection of unknown attacks and UNSW-NB15_full-test dataset to evaluate effectiveness in detection of known and unknown attacks. We find that transfer learning and fine-tuning improve the model for the detection of in detection of known and unknown attacks. The experimental results show that the transfer learning-based model achieves high accuracy, precision, F1-score and MCC. For future research, greater efforts will be made to explore and implement advanced architectures of transfer learning-based models in addressing attack detection issues by utilizing source and target domains that have different class labels, in order to further enhance the performance and feasibility of IoT attack detection systems.

FUNDING INFORMATION

Authors state no external funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Riki Abdillah Hasanuddin	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
Muhammad Subali		✓			✓	✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va: **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Publicly available dataset, model, and notebook were used in this research can be found at: https://github.com/Bedil09/Attack-detection-in-IoT-networks-with-deep-learning-using-DTL-method.git

REFERENCES

[1] T. Alam, "Internet of things: review, architecture and applications," *Computer Science and Information Technologies*, vol. 3, no. 1, pp. 31–38, 2022, doi: 10.11591/csit.v3i1.p31-38.
[2] S. Chesney, K. Roy, and S. Khorsandroo, "Machine learning algorithms for preventing IoT cybersecurity attacks," in *Intelligent Systems and Applications*, Cham: Springer, doi: 10.1007/978-3-030-55190-2_53.
[3] G. Sun, L. Liang, T. Chen, F. Xiao, and F. Lang, "Network traffic classification based on transfer learning," *Computers and Electrical Engineering*, vol. 69, pp. 920–927, 2018, doi: 10.1016/j.compeleceng.2018.03.005.
[4] J. Zhao, S. Shetty, J. W. Pan, C. Kamhoua, and K. Kwiat, "Transfer learning for detecting unknown network attacks," *Eurasip Journal on Information Security*, vol. 2019, no. 1, 2019, doi: 10.1186/s13635-019-0084-4.
[5] H. Yang and F. Wang, "Wireless network intrusion detection based on improved convolutional neural network," *IEEE Access*, vol. 7, pp. 64366–64374, 2019, doi: 10.1109/ACCESS.2019.2917299.
[6] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Deep transfer learning for IoT attack detection," *IEEE Access*, vol. 8, pp. 107335–107344, 2020, doi: 10.1109/ACCESS.2020.3000476.
[7] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," *IEEE Access*, vol. 9, pp. 103906–103926, 2021, doi: 10.1109/ACCESS.2021.3094024.
[8] Z. Ahmad *et al.*, "Anomaly detection using deep neural network for IoT architecture," *Applied Sciences*, vol. 11, no. 15, 2021, doi: 10.3390/app11157050.
[9] I. Idrissi, M. Azizi, and O. Moussaoui, "Accelerating the update of a DL-based IDS for IoT using deep transfer learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1059–1067, 2021, doi: 10.11591/ijeecs.v23.i2.pp1059-1067.
[10] J. Guan, J. Cai, H. Bai, and I. You, "Deep transfer learning-based network traffic classification for scarce dataset in 5G IoT systems," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3351–3365, 2021, doi: 10.1007/s13042-021-01415-4.
[11] Y.-C. Wang, Y.-C. Houng, H.-X. Chen, and S.-M. Tseng, "Network anomaly intrusion detection based on deep learning approach," *Sensors*, vol. 23, no. 4, 2023, doi: 10.3390/s23042171.
[12] H. Kim, S. Park, H. Hong, J. Park, and S. Kim, "Transferable deep learning framework for improving the accuracy of internet of things intrusion detection," *Future Internet*, vol. 16, no. 3, 2024, doi: 10.3390/fi16030080.
[13] L. T. Rajesh, T. Das, R. M. Shukla, and S. Sengupta, "Give and take: federated transfer learning for industrial IoT network intrusion detection," in *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2023, pp. 2365–2371, doi: 10.1109/TrustCom60117.2023.00333.
[14] J. He, X. Li, Y. Chen, D. Chen, J. Guo, and Y. Zhou, "Deep transfer learning method based on 1D-CNN for bearing fault diagnosis," *Shock and Vibration*, vol. 2021, no. 1, 2021, doi: 10.1155/2021/6687331.
[15] S. Naseer *et al.*, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.
[16] O. D. Okey, D. C. Melgarejo, M. Saadi, R. L. Rosa, J. H. Kleinschmidt and D. Z. Rodríguez, "Transfer learning approach to IDS on cloud IoT devices using optimized CNN," *IEEE Access*, vol. 11, pp. 1023-1038, 2023, doi: 10.1109/ACCESS.2022.3233775.
[17] A. I. Ahmed, Y. Etiabi, A. W. Azim and E. M. Amhoud, "A unified deep transfer learning model for accurate IoT localization in diverse environments," *2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Valencia, Spain, 2024, pp. 1-6, doi: 10.1109/PIMRC59610.2024.10817295.

[18] M. Zhong, Y. Zhou, G. Chen, "Sequential model based intrusion detection system for IoT servers using deep learning methods," *Sensors*, vol. 21, no. 4, doi: 10.3390/s21041113.
[19] E. Rodríguez *et al.*, "Transfer-learning-based intrusion detection framework in IoT networks," *Sensors*, vol. 22, no. 15, 2022, doi: 10.3390/s22155621.
[20] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, Cham: Springer, 2018, pp. 270–279, doi: 10.1007/978-3-030-01424-7_27.
[21] K. K. Patel and S. M. Patel, "Internet of things-IoT: definition, characteristics, architecture, enabling technologies, application & future challenges," *International Journal of Engineering Science and Computing*, vol. 6, no. 5, pp. 6122–6131, 2016, doi: 10.4010/2016.1482.
[22] N. Mishra and S. Pandya, "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: a systematic review," *IEEE Access*, vol. 9, pp. 59353–59377, 2021, doi: 10.1109/ACCESS.2021.3073408.
[23] T. Perumal, N. Mustapha, R. Mohamed, and F. M. Shiri, "A comprehensive overview and comparative analysis on deep learning models," *Journal on Artificial Intelligence*, vol. 6, no. 1, pp. 301–360, 2024, doi: 10.32604/jai.2024.054314.
[24] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.
[25] N. Moustafa, "The Bot-IoT dataset," *UNSW Canberra*. Accessed: Jun. 01, 2024. [Online]. Available: https://research.unsw.edu.au/projects/bot-iot-dataset
[26] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, IEEE, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
[27] N. Moustafa, "The UNSW-NB15 dataset," *UNSW Canberra*. Accessed: Jul. 01, 2024. [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset
[28] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," *Journal of Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00892-y.

BIOGRAPHIES OF AUTHORS

Riki Abdillah Hasanuddin graduated bachleor degree in Information System from Gunadarma University. His research interests include deep learning, machine learning, cloud computing and IoT. He can be contacted at email: riki.abdillah@outlook.com.

Muhammad Subali is working as Head Lecturer in the Faculty of Informatic Engineering at Cendekia Abditama University, Tangerang, Banten, Indonesia. He completed the doctoral program in 2007. His research interests include image processing, speech and signal processing. He can be contacted at email: subali@uca.ac.id.