Transfer Learning-Driven Intrusion Detection for Internet of Vehicles (IoV)

Yazan Otoum, Yue Wan, Amiya Nayak
School of Electrical Engineering and Computer Science,
University of Ottawa, Canada
Email:{yazan.otoum, ywan037, amiya.nayak}@uottawa.ca

2022 International Wireless Communications and Mobile Computing (IWCMC) | 978-1-6654-6749-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWCMC55113.2022.9825115

***Abstract*—The Internet of Vehicles (IoV) is a set of connected vehicles supported with sensors, communication technologies, and software connected by the Internet as an infrastructure. With the evolution of 5G technology, automation, and artificial intelligence, the IoV is expected to replace traditional transportation systems in the near future. On the other hand, with this evolution, the possibility of new cyberattacks has increased. This paper proposes a security framework in which intrusion detection secures the Intra/Inter-Vehicular communications within the IoV network. The proposed framework uses multi-task transfer learning to transfer knowledge gained from two different benchmark datasets. To the best of our knowledge, this is the first work that uses transfer learning to transfer the knowledge between two different benchmark datasets. The performance of the intrusion detection engine is evaluated using two different deep learning algorithms, namely Deep Neural Network (DNN) and Convolutional Neural Network (CNN), in terms of accuracy, precision, recall and F1-score. In addition to achieving satisfying performance and reduced training/fine-tuning time for the target domains, our analysis illustrates the computational effectiveness of the proposed model by transferring the knowledge from the smaller to the larger dataset.**

***Index Terms*—Transfer Learning (TL), Internet of Vehicles (IoV), Intrusion Detection System (IDS).**

I. Introduction

The Internet of Vehicles (IoV) is a new concept that is created with connected vehicles in Vehicular Ad-hoc Networks (VANETs) [1]. Based on the 2021 Statista Report [2], the expected number of autonomous and connected vehicles in the United States will be projected up to 20.8 and 146 million respectively by 2030. This rapid growth in the connected vehicles and the heterogeneity of the network communications in the IoV will increase the challenge to secure IoV networks [3]. One of the measures used to detect the inside/outside attacks that breach through existing security measures is the IDS. There are two approaches that can be used for IDS: misuse detection or anomaly detection. The misused detection approach is used to detect known attacks using their patterns, while the anomaly detection approach is used to identify any abnormal behaviour in the network. The fact that the attackers can continuously change and improve their techniques makes it impossible to predict their malicious requests to be captured in a black-list [4]. In different domains, a large number of IDSs have been proposed [5], [6], [7], [8]. Machine learning (ML) has been applied to many aspects of IoV such as objects detection, driver monitoring, self-driving, sensor fusion (i.e. radar and LiDAR), attacks and anomaly detection. ML has been used to detect attacks and maintain cybersecurity using different approaches within the IoV environments. Transfer Learning (TL) has garnered much attention as an ML technique in recent years. TL is a machine learning technique in which a model can be trained for a specific task and re-used in another related task. This technique shows that:

- It is a good solution for small training datasets, or when there is no label data, so it is computationally efficient and helps achieve better results from a small dataset.
- It can also decrease the required training time and improve the model performance compared with traditional learning methods. The models that gained knowledge, such as the features and weights from previously trained models, already understand the features, making it faster to achieve better performance than learning from scratch.

We believe that our proposed scheme is the first work to design a multitask deep transfer model using deep learning algorithms to transfer the pre-gained knowledge between two different datasets, which is the main contribution of our paper. The advantages of TL techniques motivated us to design a scalable model out of different aggregated models and overcome the limitations of the traditional ML, such as: lowering the training time, increasing the model performance, and decreasing the memory usage.

The balance of the paper is structured as follows. Section II highlights the latest research in IoV using machine and deep learning. A background on transfer learning is provided in Section III. Model workflow, datasets, and training steps have been described in Section IV. Section V illustrates the performance of the model. We conclude the paper with suggestions for potential future work in Section VI.

II. State of the Art

According to [9], the authors were able to enhance the detection rate of the IoV intrusion detection system by 23% using the transfer learning technique, based on whether or not the IoV cloud model contributes partially to the labelled data to the connected vehicles during the model update process. Upon considering the two methods for the model updating, a cloud-assisted update scheme and a local model update scheme, the connected vehicles in the proposed model were able to detect the new attacks autonomously without the help of the cloud model. For detecting different types of cyberattacks, the authors in [10] used a mixture of SMOTE oversampling algorithms and tree-based averaging feature selection methods. The authors used the CAN intrusion and CICIDS2017 datasets and achieved good accuracy and computation time.

The work [11] designed a transfer learning-based IDS that used the Convolutional LSTM to detect new attacks by

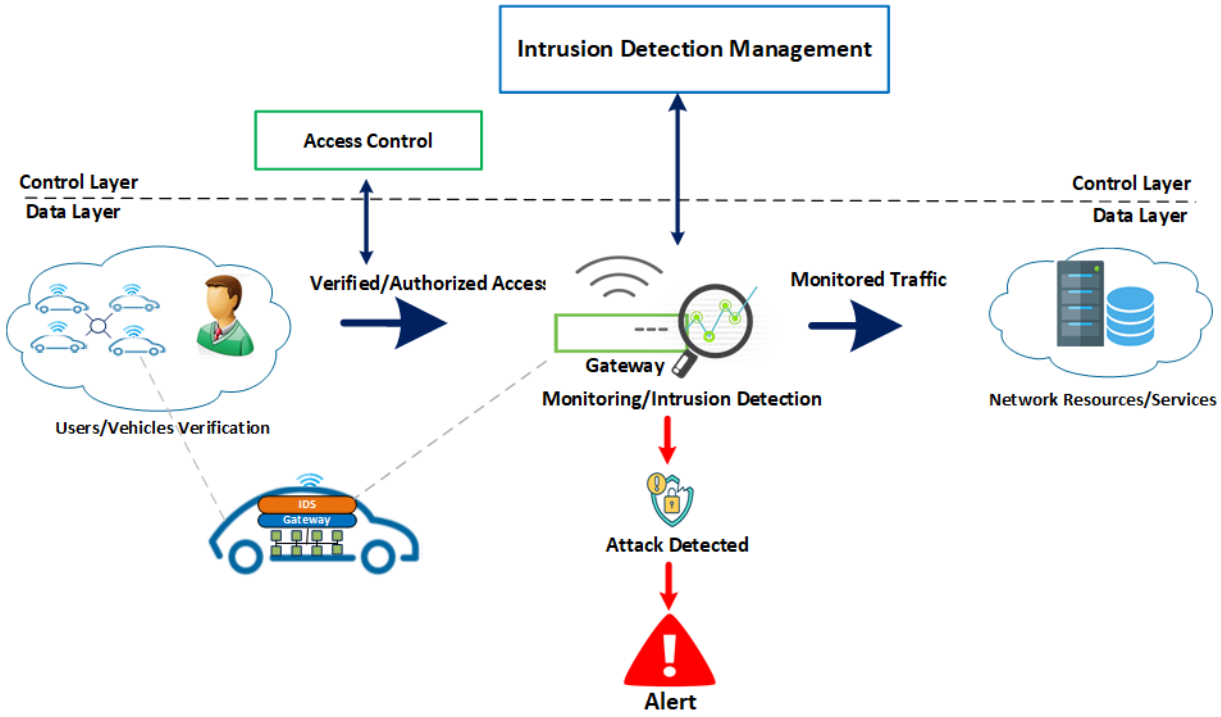

Fig. 1: Logical View of the Proposed Model

applying one-shot learning. The proposed model achieved 26.60% better performance than the other baseline machine learning algorithms using the CAN real traffic dataset. A model for detecting malicious attacks in the IoV environment is created using oversampling, outlier detection, and metric learning by the authors in [12]. The proposed approach used genetic algorithm to extract the optimal subset of features. The experimental results on the UNSW-NB15 dataset show that the proposed method achieved 98.51%, and 0.82% of accuracy and False Alarm Rate (FAR), respectively.

Transfer Learning has also been used with the help of deep learning to detect attacks in IoT networks. In their paper [13], Using two AutoEncoders (AEs), the authors present a technique for deep transfer learning (DTL). This technique allows users to learn from IoT data collected from multiple devices, not all of which are labelled. One of the two AutoEncoders (AEs) is used to train the source datasets (source domains) and another for training the target datasets (target domains). Experimental results of their model show better detection accuracy in detecting IoT attacks than other standard deep learning approaches and two recent methods.

In [14], the authors proposed a model for distributing trust across multiple policy decision points. The paper's authors studied various threshold signature schemes. They identified an appropriate scheme for the policy decision points distribution to reduce the latency and increase the model performance as much as possible. As part of another work, [15], the authors developed a policy enforcement framework to overcome some of the current challenges of the network risk-based access control. They developed a policy language to support this framework, which includes a generic firewall rule language and a mechanism to map these rules to specific firewall syntax, so they can be implemented on a firewall.

III. Background

Transfer learning is when previously learned knowledge such as features and weights is reused for a new related problem. Transfer of trained models from one domain to another involves moving models that were trained on one dataset to another dataset. If we have a domain D defined by the features space x and marginal probability $P(X)$, where X is a sample data point. Thus, $D = \{\chi, P(X)\}$, where a task, T, can be represented as a two-element tuple of the label space, Y and objective function n that can be expressed as $P(\gamma, X)$ in probabilistic terms. For a given domain D, a task is defined by $T = \{\gamma, P(Y|X)\} = \{\gamma, n\}$ and $Y = \{y_1, ..., y_n\}$, $y_i \in \gamma$. Therefore, transfer learning can be expressed for a given source domain D_s, a corresponding source task T_s, a target domain D_T, and a target task T_T. Consequently, by combining the information about the target conditional probability distribution $P(Y_T|X_T)$ in D_T with information about D_S and T_S, where $D_S \neq D_T$ or $T_s \neq T_T$, we can ensure transfer learning.

IV. The Proposed Model

A logical overview of the proposed model, illustrating the placement of the IDS in connected vehicles, is shown in Figure 1. After the detection engine screens the traffic, the security management system administrator and the driver are alerted if any attacks are detected. The proposed model passed through the following steps:

TABLE I: CICIDS2017 and CSE-CIC-IDS2018

CSE-CIC-IDS2018 Sub-dataset	Corresponding CICIDS2017 Sub-dataset	CSE-CIC-IDS2018 Attacks	Corresponding CICIDS2017 Attacks
CSE-CIC-IDS2018-Feb 14	CICIDS2017-Tuesday	FTP-BruteForce	FTP-Patator
		SSH-Bruteforce	SSH-Patator
CSE-CIC-IDS2018-Feb 15,16	CICIDS2017-Wednesday	DoS GoldenEye	DoS-GoldenEye
		DoS slowloris	DoS slowloris
		DoS Slowhttptest	DoS-SlowHTTPTest
		DoS Hulk	DoS Hulk
CSE-CIC-IDS2018-Feb 22,23,28, Mar 1	CICIDS2017-Thursday	Web Attack - Brute-force	Brute-force -Web
		Web Attack – XSS	Brute-force -XSS
		Web Attack – Sql Injection	SQL Injection
		Infiltration	Infiltration
CSE-CIC-IDS2018-Feb 20,21, Mar 2	CICIDS2017-Friday	Botnet	Botnet
		DDoS LOIT	DDoS attacks-LOIC-HTTP
		DDoS-LOIC-UDP	N/A
		DDOS-HOIC	N/A
N/A	CICIDS2017-Friday	N/A	Port Scan
N/A	CICIDS2017-Wednesday	N/A	Heartbleed Attack

1) Preprocessing: Perform the prepossessing for both datasets, which include the following; mapping the attacks labels in both CICIDS2017 and CSE-CIC-IDS2018 datasets, Table I shows the corresponding attacks in each dataset—and mapping the features of the two datasets. CICIDS2017 and CSE-CIC-IDS2018 datasets have 83 and 79 features, respectively, where the four extra features (Flow ID, Source IP, Source Port, and Destination IP) in the CICIDS2017 can be dropped since they have minor importance.

TABLE II: Normal traffic and attacks distributions in CSE-CIC-IDS2018 and CICIDS2017 datasets

Instance	CICIDS2017	CSE-CIC-IDS2018
Normal Traffic	1,743,179	6,112,151
DDoS Attack	128,027	687,742
DoS Attacks	252,661	654,301
Botnet Attacks	1966	286,191
Brute-force Attacks	13,835	380,949
Infiltration Attacks	36	161,934
Web Attacks Attacks	2180	928
Port Scan Attacks	158,930	-

2) Model Construction: The Deep Neural Network (DNN) and Convolutional Neural Network (CNN) algorithms were used to build the Global models using the CICIDS2017 dataset as the source domain. By analyzing the two datasets, we uncovered that using the CICIDS2017 dataset as the source domain needs less training time thanks to fewer samples and gives better performance, as discussed later in Section V of this paper. A feature representation that maps the original feature space into a shared subspace between two or more datasets can be used to transfer knowledge using Neural Networks. By doing so, insufficient data instances will be dealt with, and training time will be reduced—consequently, The more similar the domains for which the TL can be used, the better the model will perform. The DNN network has six layers: one input layer, four hidden layers of 32, 16, 8, and 4 nodes, and a Softmax layer which serves as a classification layer. On the other hand, the CNN has eight layers: one input, one Convolutional1D, one Max pooling1D, one Flattens layer, one Dropout, two dense layers, and an output layer. Using the DNN and CNN algorithms, the global model is trained on the CICIDS2017 dataset and then assigned to the local models on the edges. With edge datasets, the local models are then trained using cross-entropy loss based on the following equation:

$$L(\Theta) = -\sum_{i=1}^{k} y_i \log(\hat{y}_i) \tag{1}$$

where the number of input training samples is i, and the value of the label is y. After training the local datasets with the global models, the new aggregated model will be composed without disclosing the user's personal data, based on the parameters of the newly trained models; a similar approach is applied in [16]. In order to calculate an aggregated model, the local models are averaged, and it is represented as follows:

$$M_g(x) = \frac{1}{N}\sum_{k=1}^{N} M_{lk}(x) \tag{2}$$

where N is the number of local models and x is the parameter set. Edge datasets can directly be assigned to the aggregated model. However, it will not perform well because it learns only the shared features across all datasets and not those customized for each dataset. In this case, a more efficient way to do transfer learning is to freeze all layers, replace

the output layer with a new output layer, and then unfreeze the layers individually according to the performance desired. During the backpropagation process, the model freezes the first few layers that hold the low-level features from each edge dataset and tries to learn the more specific features from the other layers. Adding a new edge dataset will allow a more efficient generalization of the global model. Furthermore, the global model can be updated through synchronization with the edge models and passing the updated models to all edge nodes to apply transfer learning and create a new personalized local model.

3) Layers Freezing Technique: One of the techniques used to enhance the model performance and decrease the training time in the transfer learning is the layers freezing in which we freeze the gained parameters of some of the top layers of the pre-trained model that have the generic features and only update (fine-tune) the other layers' parameters that have the specific high-level features. Our experiments show the effectiveness of freezing the layers for two different cases: freezing the first layer and freezing the first two layers. More details of the experiment for the layers freezing are shown in Section V.

4) Knowledge Transfer: In this step, the knowledge gained from the previous step will be transferred and reused to build a new local model with the CSE-CIC-IDS2018 sub-datasets in the same way as the training algorithms mentioned in the previous step. The Inductive Multi-Task Transfer learning approach has been used since the source and the target domains (features) are the same. Multi-task learning is a special type of multi-output learning [17] that learns different related tasks with different training datasets or features at the same time with the aim to enhance the generalization performance or increase the relationship between those tasks. However, multi-output learning uses the same training set. The newly trained local models are transferred back to be aggregated to build a new global model using the transferred parameters from the local models. The used datasets share the same feature spaces but differ in the instances of source and target task labels (attacks). As a result, each local edge uses a customized model constructed by combining the aggregated global model with the local data. So, the best possible performance is achieved while maintaining data privacy (closest to that obtained by directly training the model).

5) Datasets: CSE-CIC-IDS2018 dataset [18] contains about 16 million instances collected through different ten days. CSE-CIC-IDS2018 has 80 network traffic features and seven main types of attacks: DDoS [19], DoS, Botnet, Brute-force, Infiltration, Web attack, and Botnet attack. These represent more than 60% of the total number of instances. As for the CICIDS2017 dataset [20], which contains packet captures of real-time network traffic generated by the CICFlowMeter traffic generator, it has real-time packet captures as well as network traffic. On Monday, Tuesday, Wednesday, Thursday, and Friday, various common attacks were detected, including Brute-force, Heartbleed, botnet, DoS, DDoS, web, and infiltration attacks, but only normal traffic was detected on Monday. Table II shows the number of normal traffic and attacks instances in both CSE-CIC-IDS2018 and CICIDS2017 datasets.

V. Performance Evaluation

To evaluate the performance of the detection engine, we consider the performance metrics accuracy, precision, recall, F1-score, and the average of training time and fine tuning time (The time needed to tune the model for the new task). The dataset CICIDS2017 was chosen to train the global model using the DNN and CNN algorithms for two main reasons. First, this dataset has fewer than 3 million instances compared to the CSE-CICIDS2018 that has more than 16 million instances, which also illustrates the effectiveness of the model in transferring the gained knowledge from the relatively small dataset to the large dataset. Second, as shown Figure 2 and Figure 3, we found by the experiments that using CICIDS2017 to train the global model gives a better performance for the following two cases.The first case is shown in Figure 2 where the first layer is frozen, and the other layers' parameters were tuned. The second case shown in Figure 3 shows the effects of freezing the top two hidden layers and tuning the parameters in the other hidden layers. In both cases, we took the average performance for the selected attacks (DoS, DDoS, Botnet, and Brute-force).

In Figure 4, we show the average performance of freezing the top three layers of the DNN and CNN algorithms when using both datasets. The performance contrast between the two algorithms when freezing different layers results from the fact that DNN and CNN algorithms have 9617 and 48951 trainable parameters, respectively. Those parameters will be decreased when freezing some of the layers and after fine-tuning the parameters in the remaining layers. Based on the experiments, we conclude that DNN works better using 177 trainable parameters when freezing the top two layers by achieving 97.83%, and 99.97% accuracy when using CSE-CICIDS2018 and CICIDS2017 datasets as the global model, respectively. On the other hand, CNN works better using 279 trainable parameters that satisfied with freezing the first layer by achieving 97.96%, and 99.90% accuracy when using CSE-CICIDS2018 and CICIDS2017 datasets as the global model, respectively. However, the minimum required trainable parameters to get a satisfactory performance will be 41 and 39 for DNN and CNN, respectively. To support this argument we calculate the average training and fine tuning times for both algorithms in Table III, which shows how the time needed to customize the model is significantly less than training it directly. Performance and training time must be balanced to maximize the model's performance; a better model performance needs more trainable parameters and takes more training time. However, those times could also be affected by machine capabilities. We implemented the algorithms using Keras/Pytorch running as a Python library on TensorFlow cloud servers and a workstation with Intel Core i7 processor and 16GB DDR4 RAM. We utilized the Google Collaboratory cloud servers and a computer with Intel Core i7 processor.

TABLE III: Training and Fine Tuning Time Comparison

	Algorithms			
	DNN		CNN	
Attacks	**Avg. Training Time (S)**	**Avg. Fine Tuning Time (S)**	**Avg. Training Time (S)**	**Avg. Fine Tuning Time (S)**
DoS	732.56	595.09	826.41	647.83
DDoS	916.65	894.79	854.35	789.90
Botnet	368.05	164.21	313.44	236.83
Brute-force	223.71	114.29	312.81	239.60

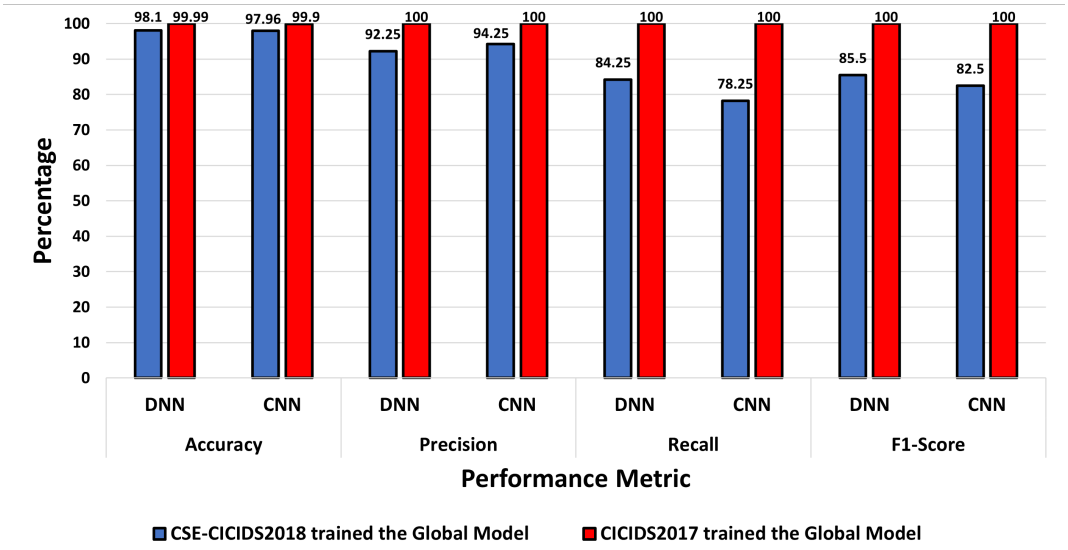

Fig. 2: Model performance for all attacks when freezing first layer

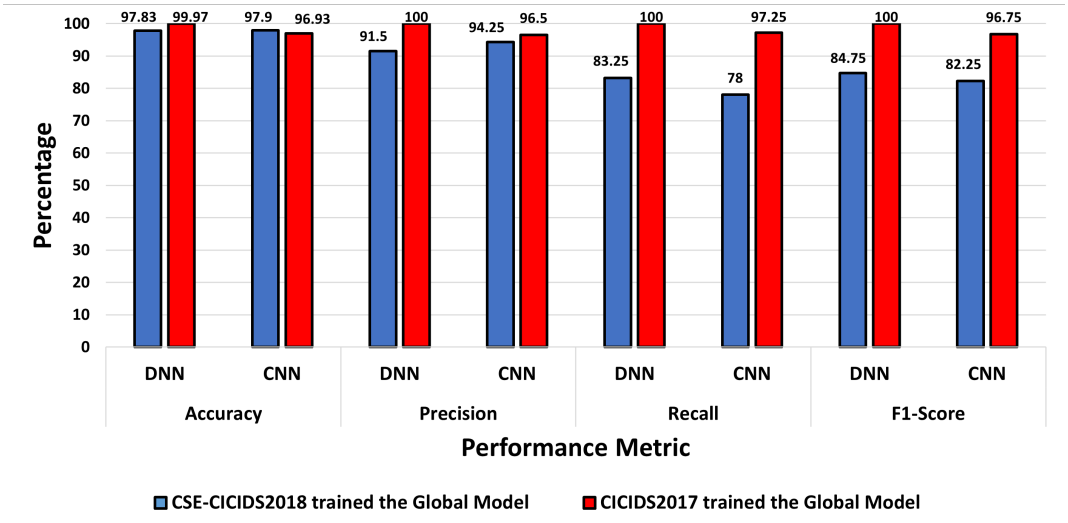

Fig. 3: Model performance for all attacks when freezing top two layers

VI. Conclusion

This paper presents a security framework that uses the transfer learning technique to secure Internet of Vehicle (IoV) networks. Using DNN and CNN algorithms, the model trains the cloud model by transferring the knowledge from the CICIDS2017 dataset to the CSE-CIC-IDS2018 dataset. Our findings show that DNN works best with 177 trainable parameters that achieve satisfactory performance by freezing two layers. In comparison, CNN works best with 279 trainable parameters that satisfy the first layer's freezing. We examine

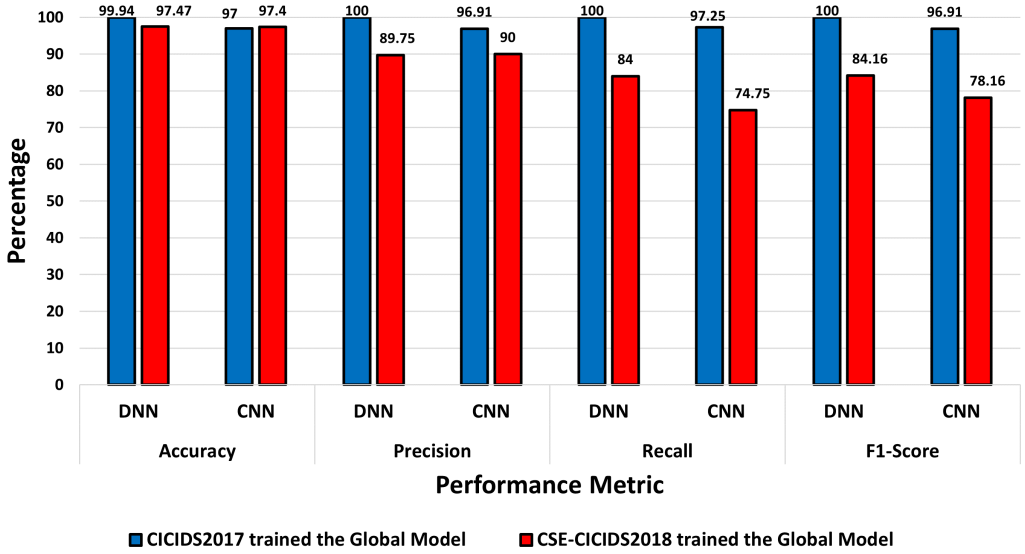

Fig. 4: Model average performance for all attacks

performance metrics, such as accuracy, precision, recall, F1-score, and the average of training and fine-tuning times. The proposed model successfully transfers knowledge from the relatively smaller dataset to the larger dataset, according to the results. In the future, the model's performance can be investigated for other types of attacks and using other deep learning algorithms such as Recurrent Neural Network (RNN).

REFERENCES

[1] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: architecture, protocols, and security," *IEEE internet of things Journal*, vol. 5, no. 5, pp. 3701–3709, 2017.

[2] M. Placek, "U.s. - connected vehicles 2030," Aug 2021. [Online]. Available: https://www.statista.com/statistics/750113/us-connected-vehicles/

[3] D. S. Gupta, A. Karati, W. Saad, and D. B. Da Costa, "Quantum-defended blockchain-assisted data authentication protocol for internet of vehicles," *IEEE Transactions on Vehicular Technology*, 2022.

[4] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," *IEEE Access*, vol. 8, pp. 106 576–106 584, 2020.

[5] Y. Otoum and A. Nayak, "Signature-over-the-air with transfer learning ids for intelligent connected vehicles (icv)," in *2021 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2021, pp. 1–6.

[6] T. Alladi, V. Kohli, V. Chamola, F. R. Yu, and M. Guizani, "Artificial intelligence (ai)-empowered intrusion detection architecture for the internet of vehicles," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 144–149, 2021.

[7] S. Otoum, B. Kantarci, and H. Mouftah, "A comparative study of ai-based intrusion detection techniques in critical infrastructures," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 4, pp. 1–22, 2021.

[8] Y. Otoum, Y. Wan, and A. Nayak, "Federated transfer learning-based ids for the internet of medical things (iomt)," in *2021 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2021, pp. 1–6.

[9] X. Li, Z. Hu, M. Xu, Y. Wang, and J. Ma, "Transfer learning based intrusion detection scheme for internet of vehicles," *Information Sciences*, vol. 547, pp. 119–135, 2021.

[10] L. Yang, A. Moubayed, I. Hamieh, and A. Shami, "Tree-based intelligent intrusion detection system in internet of vehicles," in *2019 IEEE global communications conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[11] S. Tariq, S. Lee, and S. S. Woo, "Cantransfer: transfer learning based intrusion detection on a controller area network using convolutional lstm network," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 1048–1055.

[12] F. Jin, M. Chen, W. Zhang, Y. Yuan, and S. Wang, "Intrusion detection on internet of vehicles via combining log-ratio oversampling, outlier detection and metric learning," *Information Sciences*, vol. 579, pp. 814–831, 2021.

[13] 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. 107 335–107 344, 2020.

[14] B. Sengupta and A. Lakshminarayanan, "Distritrust: Distributed and low-latency access validation in zero-trust architecture," *Journal of Information Security and Applications*, vol. 63, p. 103023, 2021.

[15] R. Vanickis, P. Jacob, S. Dehghanzadeh, and B. Lee, "Access control policy enforcement for zero-trust-networking," in *2018 29th Irish Signals and Systems Conference (ISSC)*. IEEE, 2018, pp. 1–6.

[16] X. Li, H.-l. Chi, W. Lu, F. Xue, J. Zeng, and C. Z. Li, "Federated transfer learning enabled smart work packaging for preserving personal image information of construction worker," *Automation in Construction*, vol. 128, p. 103738, 2021.

[17] D. Xu, Y. Shi, I. W. Tsang, Y.-S. Ong, C. Gong, and X. Shen, "Survey on multi-output learning," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2409–2429, 2019.

[18] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data," *Journal of Big Data*, vol. 7, no. 1, pp. 1–19, 2020.

[19] Y. Chen, S. Das, P. Dhar, A. El-Saddik, and A. Nayak, "Detecting and preventing ip-spoofed distributed dos attacks." *Int. J. Netw. Secur.*, vol. 7, no. 1, pp. 69–80, 2008.

[20] R. Panigrahi and S. Borah, "A detailed analysis of cicids2017 dataset for designing intrusion detection systems," *International Journal of Engineering & Technology*, vol. 7, no. 3.24, pp. 479–482, 2018.