Intrusion Detection System for Internet of Vehicles using Optimized CNN

Prof. Prakash M
dept. School of Advance Science
Vellore Institute of Techonology
Vellore, India.
prakash.m@vit.ac.in

Kishorkumar K R
dept. School of Advance Science
Vellore Institute of Techonology
Vellore, India.
kishorkumar.offl@gmail.com

Santhosh kumar S
dept. School of Advance Science
Vellore Institute of Technology
Vellore, India.
ssanthoshkumar.official@gmail.com

***Abstract*—In this modern world, there are many new technologies evolving and in the same way Cyber-attack is also following up. Modern vehicles, including Electric vehicles, autonomous vehicles and connected vehicles, are increasingly connected to the external world, which enables various functionalities and services. These inbuild technologies in vehicles helps the cyber attackers to attack the Internet of Vehicles (IoV), causing its vulnerabilities to cyber threats. Due to the need of more security and encryption performance in vehicular network, Intrusion Detection Systems (IDSs) are much needed strategy to protect the modern vehicle systems from cyber threats. In this paper, we are referring the Transfer Learning with use of ML and ensemble learning-based IDS is proposed for IoV using Recurrent Neural Network (RNN) and some hyper-parameter optimization techniques. The Car-Hacking dataset and the CICIDS2017 dataset, we are using here. This will show the definitiveness of the proposed IDS for Cyber-attack detection in both intra and external vehicular networks.**

***Index Terms*—Intrusion Detection System, CNN, Transfer learning, Ensemble learning, Internet of Vehicles.**

I. Introduction

Modern automobiles have evolved into network-controlled vehicles due to the Internet of Things' (IoT) and Internet of Vehicles' (IoV) technologies' rapid development, including connected and autonomous vehicles (AVs and CVs). Most IoV systems use intra-vehicle communications. Both internal and external networks (IVNs). The Controller in IVNs the main system that makes this possible is the Area Network (CAN) bus. electronic control units (ECUs) to communicate with one another implement functionalities and take actions. As opposed to that, Networks of external vehicles provide communication between smart cars and other IoV elements, such as roadside units, constructions, and users of roads. [1] However, the expanded cyber-attack surfaces of contemporary automobiles have resulted from the increased identity and accessibility of automotive networks. Additionally, since CAN packets are of a certain length, processing these packets does not employ any authentication or encryption techniques. Cyber attackers can conduct several types of attacks, including DoS, fuzzy, and spoofing attacks, by injecting malicious messages into IVNs due to the absence of fundamental security safeguards. On the other hand, because of the developing cellular connections between connected vehicles and external networks, these vehicles are becoming more susceptible to common cyberattacks. In order to safeguard IoV systems and smart vehicles by recognizing cyber-attacks, it is essential to build intrusion detection systems (IDSs). [2] Researchers and automakers have recently become interested in machine learning (ML) and deep learning (DL) applications in cyber security and car systems due to the advancement of these techniques. In order to create classifier based IDSs that can differentiate between regular network traffic and other cyber-attacks through traffic data analytics, ML and DL techniques are frequently applied. This research suggests an intelligent IDS model to safeguard IoV systems based on enhanced Convolutional Neural Networks (CNNs), transfer learning, and ensemble learning methods. Because of car network traffic data, base learners are trained using VGG16, VGG19, Xception, Inception, and InceptionResnet, five cutting-edge CNN models. [1] Particle Swarm Optimization (PSO), a hyper-parameter optimization (HPO) technique, is used to tune the hyper-parameters of the CNN models to provide optimum learning models. To further enhance the intrusion detection performance, the underlying CNN models are then combined by using the ensemble procedures of confidence averaging and concatenation. The Car-Hacking dataset and the CICIDS2017 dataset are two open-source vehicle network datasets that are used to assess the efficacy and efficiency of the proposed IDS system. The primary contributions made in this study are as follows:

1) Through CNN, transfer learning, ensemble learning, and HPO approaches, it presents a unique framework for efficient cyber-attack detection in both internal and external networks.

2) It suggests a data transformation technique that can successfully convert data from car network traffic into visuals, making it simpler to identify different cyber-attack patterns.

3) It compares the performance of the proposed method to other state-of-the-art methods approaches using two benchmark cyber-security datasets that reflect data from internal and external networks.

II. Related Work

IoV intrusion detection tasks have seen widespread use of ML and DL models. A Multi-Layer Perceptron-based DL-based IDS for connected automobiles was proposed by Rosay et al. (MLP) Using the CICIDS2017 dataset, the MLP model

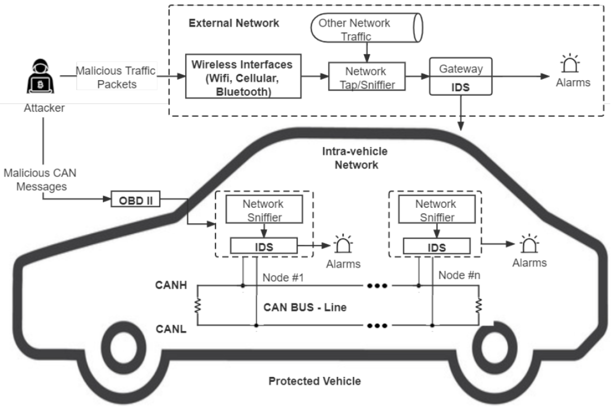


Fig. 1. Architecture of IDS-Protected Vehicle

was assessed on an automobile microprocessor. A tree-based stacking approach was put out by Yang et al. For the analysis of network traffic in IoV situations. The IoV and CICIDS2017 datasets demonstrate the great effectiveness of the stacking strategy. The development of CNN-based IDS for automotive networks was the subject of several earlier works. Mehedi et al. introduced the P LeNet approach for deep transfer learning-based in-vehicle network intrusion detection. On the Car-Hacking dataset, the P-LeNet model received a high F1-score of 97.83 percentage. For intra-vehicle intrusion detection, Hossain et al. developed a one-dimensional CNN (1D-CNN) based IDS since 1D-CNN models perform well in numerous time-series data analytics challenges. In order to identify attacks on IVNs, Song et al. suggested an IDS model based on deep CNN (DCNN) and employing reduced InceptionResnet. The Car-Hacking dataset demonstrates the great accuracy of the DCNN model. [3] Although the above methods achieve high accuracy in IoV cyber-attack detection tasks, there is still much room for performance improvement. The proposed solution aims to construct an optimal IDS framework using state-of-the-art CNN models optimized using HPO and ensemble learning strategies. Additionally, transfer learning techniques are used to improve the model training efficiency.

III. Proposed Framework

This effort aims to defend both internal and external vehicular networks by creating an IDS that can identify different forms of threats. The design of an IDS-protected vehicle and the typical attack scenario. By transmitting malicious traffic packets, cybercriminals can launch internal attacks on IVNs using the On-Board Diagnostics II (OBD II) interface and external assaults on external vehicular networks via wireless interfaces. The proposed IDS should therefore be implemented in both IVNs and outside networks. The suggested IDS can be installed in IVNs on top of the CAN-bus to identify unusual CAN messages and produce alarms. The proposed IDS can be integrated into gateways in external networks to detect and filter all malicious packets with the intention of breachthe vehicles. In order to identify different forms of assaults in IoV systems, an unique optimized CNN and transfer learning based IDS is proposed in this research. exemplifies the proposed IDS framework's overview. The quantile transform method is first used to convert the time-based chunked intra-vehicle and external network data into images. [5] after that In order to build basic learners, the generated image collection is trained using five cutting-edge CNN models (VGG16, VGG19, Xception, Inception, and InceptionResnet). PSO, an HPO technique that can automatically tune the hyper-parameters, is used to optimize the CNN models. The top three performing CNN models are then chosen as the initial three CNN models. Build the models for ensemble learning. Two ensemble techniques, confidence averaging and concatenation, are used to construct ensemble models for final detection.

IV. Data Description and Transformation

In this work, two datasets are used to create the proposed IDS for both IVNs and external vehicular networks. The Car-Hacking dataset, which constitutes the first dataset, [6] intra-vehicle data, as it is produced by sending CAN packets into an actual vehicle's CAN-bus. The primary components of the dataset are the CAN identification (ID) and the 8-bit data field of CAN packets. Attacks of the DoS, fuzzy, gear spoofing, and Revolutions Per Minute (RPM) spoofing varieties are the four main categories included in the Car-Hacking dataset. The CICIDS2017 dataset, which represents external network data, is the second dataset selected since it is a cutting-edge network security dataset that contains the most recent attack patterns. The CICIDS2017 attack patterns are depicted in dataset analysis, which claims that DoS assaults, port-scan attacks, brute-force attacks, web attacks, and botnets are the five main types of attacks that may be categorised in the dataset. The data must be pre-processed after acquisition in order to produce an appropriate input for the suggested IDS. The original network data should be converted into image forms since CNN models perform better on image sets while datasets for vehicular network traffic are often tabular data. Data normalisation is the first step in the data transformation process. [7] The network data should also be standardised into the scale of 0-255 since the pixel values of images range from 0 to 255. The two most popular techniques for normalising data into the same range are quantile normalisation and minmax normalisation.

Quantile normalisation is utilised in the suggested architecture since min-max normalisation struggles with outliers and may result in the majority of data samples having incredibly low values. The quantile normalisation approach recalculates all of the feature values based on the normal distribution after converting the feature distribution to a normal distribution. As a result, the bulk of variable values are within a few standard deviations of the median, effectively handling outliers. Using the timestamps and feature sizes of network traffic datasets, the data samples are divided into chunks after normalisation. The data samples are transformed following data normalisation. depending on the timestamps and size of the features of datasets for network traffic. As it has done for the Car-Hacking dataset 9 key characteristics (CAN ID and DATA[0]–DATA[7]), each

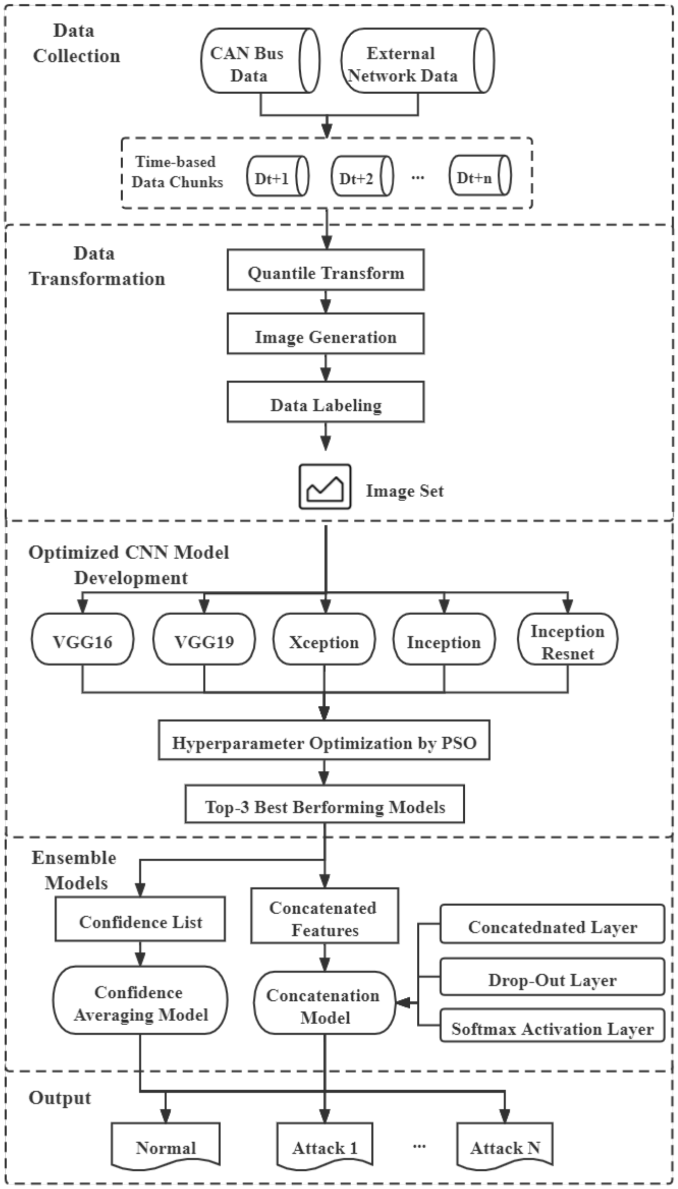
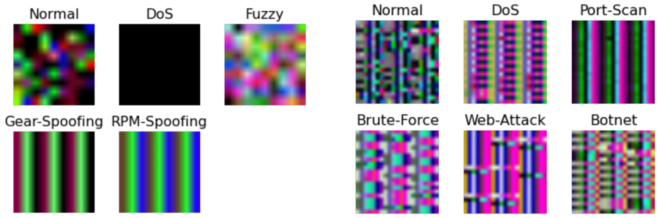


Fig. 2. CNN based IDS-Framework

of which block of 9 characteristics and 27 consecutive samples (279 = 243) [8] feature values combined) are converted into a shape picture. 9x9x3 [14]. Each modified picture is a square colour as a result. [9] a three-channel picture (red, green, and blue). Similarly, the CICIDS2017 dataset that produced 20 significant characteristics Each image from is converted to 20203 colour images. This dataset's chunk has 203 = 60 consecutive data. samples. Since the timestamps are used to produce the photos the initial time-series correlations of the data samples, network data can be retained. [**?**]



(a) Car-Hacking dataset.

(b) CICIDS2017.

Fig. 3. Sample images of each class in two datasets: a) Car-Hacking dataset; b) CICIDS2017.

V. CNN and Transfer Learning

DL models like CNN are frequently employed in picture classification and recognition issues. Without the need for additional feature extraction and data reconstruction procedures, the images can be immediately entered into CNN models. Convolutional, pooling, and fully-connected layers make up the majority of layers in a CNN. The feature patterns of images can be automatically retrieved by convolution operations in convolutional layers. Local correlations can be used in pooling layers to minimise data complexity without sacrificing crucial information and prevent over-fitting. All features are connected through fully connected layers, which also produce the output. [10] Transfer Learning (TL) is the process of moving a Deep Neural Network (DNN) model's weights from one dataset to another for DL models. [11]

Numerous picture types have successfully used the TL approach. processing activities This is due to the learnt feature patterns. Only the features learnt by the top layers of CNN models are specific characteristics for a given dataset, whereas the features learned by the bottom layers of CNN models are often generic patterns that are relevant to many different jobs . As a result, the CNN models' base layers may be used directly to various applications. Fine-tuning may be applied to the TL process of DL models to increase its efficacy. Most of the pre-trained model's layers are frozen (i.e., their weights are kept) during fine-tuning, while a few of the top layers are unfrozen to retrain the model on fresh data. [8]

We have chosen VGG16, VGG19, Xception, Inception, and InceptionResnet as the basis models for the proposed system.

Considering the effectiveness of CNN models in the majority of picture categorization issues, These CNN models have shown excellent performance on a variety of picture classification tasks after being pre-trained on the ImageNet dataset. A benchmark image processing dataset with more than a million photos in 1,000 classes is called ImageNet. On the ImageNet Challenge, the 16-layer (VGG16) and 19-layer (VGG19) VGG16 models suggested in had a decreased error rate of 7.3 percentage. While the VGG19 design adds three more convolutional layers, the VGG16 architecture consists of five blocks of convolutional layers and three fully linked layers.

VI. Proposed Ensemble Learning Model

Ensemble learning is a method that combines several using fundamental learning models to build an ensemble model increased efficiency. The use of ensemble learning is common in data analytics issues because a combination of several

Learners typically outperform solitary learners. An technique to ensemble learning called confidence averaging combines the basic learners' categorization probability values. To identify the group with the greatest confidence level. In Softmax layers in DL models can produce a posterior probability, list that includes each class's categorization confidence. [11] For each class, the confidence averaging approach determines the average classification probability of base learners. the class label with the highest average confidence is returned. As the ultimate categorization outcome.

$$\mathrm{Softmax}(\mathbf{z})_i = \frac{e^{\mathbf{z}_i}}{\sum_{j=1}^{C} e^{\mathbf{z}_j}}$$

Fig. 4. softmax formula

VII. Hyper-Parameter Optimization (HPO)

To further improve the foundation models' fit to the chosen datasets and enhance the performance of the models, and the hyper-parameters CNN models must be adjusted and improved. CNN models, like other DL models, have a substantial how many hyper-parameters need to be tuned. This set of hyperparameters can be categorised as hyper-parameters for model creation. and hyper-parameters for model training. Model-design The hyper-parameters that need to be set are hyper-parameters. when developing a model. The suggested TL structure includes The number or range of the model-design hyper-parameters can be frozen layer proportion, learning rate, and dropout rate rate. Hyper-parameters for model training, on the other hand are employed to balance model performance and training pace, an early stop, the number of epochs, and the batch size patience.

The structure, potency, and efficiency of CNN models are directly influenced by the aforementioned hyper-parameters. HPO is an automated procedure that uses optimization approaches to fine-tune the hyper-parameters of ML or DL models [10]. PSO is a popular metaheuristic optimization methodology for HPO issues that determines optimal hyper-parameter values through information exchange and teamwork among the particles in a swarm [10]. Each member of the group is initialised with a location xi and velocity vi at the beginning of PSO. Each particle's velocity is updated depending on its own most recent best position, pi, and the most recent global ideal position, p.

VIII. Performance and Evaluation

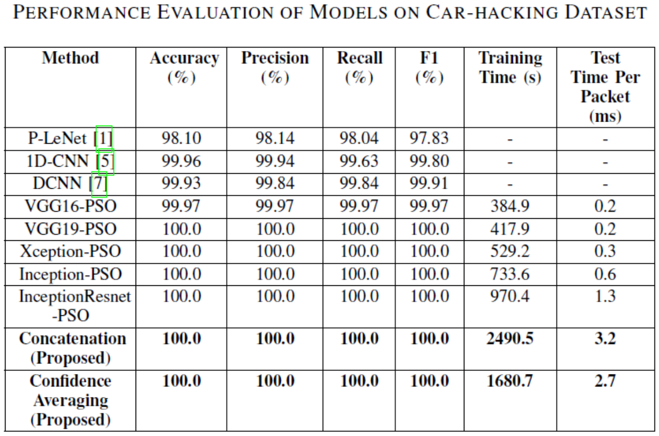
A. Experimental setup

Python libraries called Scikit-learn and Keras were used to carry out the studies. The suggested DL models were evaluated on a Raspberry Pi 3 with a BCM2837B0 64-bit CPU and 1 GB of memory and trained on a Dell Precision 3630 with an i7-8700 processor and 16 GB of memory, which represented an IoV central server machine and a vehicle-level local machine, respectively. As stated in Section III-B, the proposed architecture is tested using the benchmark CICIDS2017 and Car-Hacking datasets for vehicle network security. The suggested model is assessed using fivefold cross-validation, which can prevent biassed and over-fitted findings. [18]

On the other hand, as network traffic data is typically very unbalanced and only contains a tiny proportion of attack samples, performance is assessed using four separate metrics: accuracy, precision, recall, and F1- scores. Additionally, model training time on the server-level machine and model testing time on the vehicle-level machine are tracked and compared to assess the efficacy of the proposed strategy.

B. Experiments and Results

The primary hyper-parameters of each basic CNN model in the suggested framework were all tuned using PSO to provide the best models possible. The HPO technique was only used for the CICIDS2017 dataset because CNN models with default hyperparameter settings can already attain accuracy levels of close to 100 percentage on the Car-Hacking dataset.

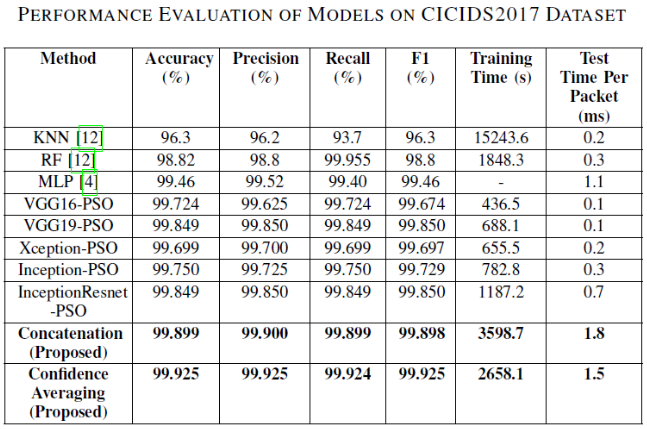


Performance Evaluation of Models on Car-hacking Dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Training Time (s)	Test Time Per Packet (ms)
P-LeNet [1]	98.10	98.14	98.04	97.83	-	-
1D-CNN [5]	99.96	99.94	99.63	99.80	-	-
DCNN [7]	99.93	99.84	99.84	99.91	-	-
VGG16-PSO	99.97	99.97	99.97	99.97	384.9	0.2
VGG19-PSO	100.0	100.0	100.0	100.0	417.9	0.2
Xception-PSO	100.0	100.0	100.0	100.0	529.2	0.3
Inception-PSO	100.0	100.0	100.0	100.0	733.6	0.6
InceptionResnet -PSO	100.0	100.0	100.0	100.0	970.4	1.3
Concatenation (Proposed)	**100.0**	**100.0**	**100.0**	**100.0**	**2490.5**	**3.2**
Confidence Averaging (Proposed)	**100.0**	**100.0**	**100.0**	**100.0**	**1680.7**	**2.7**

Fig. 5. Evaulation of models on car hacking dataset.

the starting search space and the ideal hyperparameter settings. Following HPO, the suggested ensemble models were built using the optimised CNN models as foundation learners. Tables II and III, respectively, present the findings of testing the enhanced CNN models and the suggested ensemble models on the Car-Hacking and CICIDS2017 datasets. Table II demonstrates that, with the exception of VGG16, all improved basic CNN models attain 100 percentage accuracy and F1-scores. This is mostly due to the fact that the modified pictures displayed in Fig. 3 make it easy to discern between the normal and attack patterns in the Car-Hacking dataset. Concatenation and confidence averaging procedures, two ensemble methodologies, can also Follows, data transformation and PSO, the optimised base CNN models for the CICIDS2017 dataset attain high F1-scores of 99.674 percent to 99.850 percent, as shown in fig 6. Additionally, the suggested confidence



PERFORMANCE EVALUATION OF MODELS ON CICIDS2017 DATASET

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Training Time (s)	Test Time Per Packet (ms)
KNN [12]	96.3	96.2	93.7	96.3	15243.6	0.2
RF [12]	98.82	98.8	99.955	98.8	1848.3	0.3
MLP [4]	99.46	99.52	99.40	99.46	-	1.1
VGG16-PSO	99.724	99.625	99.724	99.674	436.5	0.1
VGG19-PSO	99.849	99.850	99.849	99.850	688.1	0.1
Xception-PSO	99.699	99.700	99.699	99.697	655.5	0.2
Inception-PSO	99.750	99.725	99.750	99.729	782.8	0.3
InceptionResnet -PSO	99.849	99.850	99.849	99.850	1187.2	0.7
Concatenation (Proposed)	**99.899**	**99.900**	**99.899**	**99.898**	**3598.7**	**1.8**
Confidence Averaging (Proposed)	**99.925**	**99.925**	**99.924**	**99.925**	**2658.1**	**1.5**

Fig. 6. EVALUATION OF MODELS ON CICIDS2017 DATASET

averaging ensemble model has the greatest F1-score of 99.925 percent, just edging out the concatenation model's F1-score of 99.899 percent. Additionally, the two ensemble models perform better than other recent approaches in the literature. Additionally, the confidence averaging method requires significantly less training time overall [**?**] than the concatenation strategy. The advantages of utilising CNN, TL, and HPO approaches are supported by the suggested models' superior performance when measured against other cutting-edge IDSs.

Additionally, the average length of the proposed ensemble models' test and prediction cycles Each packet on the Raspberry Pi computer is running at a modest volume, according to fig 5 and 6, ranging from 1.5 ms to 3.2 ms. As the The demand for real-time in vehicle anomaly detection systems about 10 ms for each packet's inspection, the low The presented models' prediction times demonstrate their viability. on real-time IoV systems of implementing the suggested IDS.

IX. CONCLUSION

The cyberthreats to IoV systems are considerably growing as modern automobiles become more linked. This paper suggested a transfer learning and ensemble learning-based IDS framework that employs optimal CNN models to recognise various forms of assaults in IoV systems in order to safeguard connected cars from being infiltrated by cyber-attacks. A chunk-based data transformation technique is also suggested for converting car network traffic data into picture data for CNN models. The Car-Hacking and CICIDS2017 datasets, which contain intra-vehicle and external network data, respectively, are used to test the proposed IDS. The experimental findings demonstrate that, when compared to existing state-of-the-art approaches on the two benchmark datasets, the proposed IDS framework can more successfully identify different types of attacks with higher F1-scores of 100 percent and 99.925 percent. Additionally, the model testing outcomes on a machine at the vehicle level demonstrate the viability of the suggested IDS in real-time vehicle networks. Future research will build on this framework to create an online adaptive model that can solve concept drift in time-series car network data and accomplish online learning.

REFERENCES

[1] Liao, H. J., Lin, C. H. R., Lin, Y. C., and Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16-24.

[2] Smaha, S. E. (1988, December). Haystack: An intrusion detection system. In Fourth Aerospace Computer Security Applications Conference (Vol. 44).

[3] Vigna, G., Kemmerer, R. A. (1999). NetSTAT: A network-based intrusion detection system. Journal of computer security, 7(1), 37-71.

[4] Hoque, M. S., Mukit, M., Bikas, M., Naser, A. (2012). An implementation of intrusion detection system using genetic algorithm. arXiv preprint arXiv:1204.1336.

[5] Javaid, A., Niyaz, Q., Sun, W., Alam, M. (2016, May). A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS) (pp. 21-26).

[6] Heady, R., Luger, G., Maccabe, A., Servilla, M. (1990). The architecture of a network level intrusion detection system (No. LA-SUB-93-219). Los Alamos National Lab.(LANL), Los Alamos, NM (United States); New Mexico Univ., Albuquerque, NM (United States). Dept. of Computer Science.

[7] Ilgun, K. (1993, May). USTAT: A real-time intrusion detection system for UNIX. In Proceedings 1993 IEEE Computer Society Symposium on Research in Security and Privacy (pp. 16-28). IEEE.

[8] Peddabachigari, S., Abraham, A., Grosan, C., Thomas, J. (2007). Modeling intrusion detection system using hybrid intelligent systems. Journal of network and computer applications, 30(1), 114-132.

[9] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. Ieee Access, 7, 41525-41550.

[10] Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P. N., Kumar, V., Srivastava, J., Dokas, P. (2004). Minds-minnesota intrusion detection system. Next generation data mining, 199-218.

[11] Haq, N. F., Onik, A. R., Hridoy, M. A. K., Rafni, M., Shah, F. M., Farid, D. M. (2015). Application of machine learning approaches in intrusion detection system: a survey. IJARAI-International Journal of Advanced Research in Artificial Intelligence, 4(3), 9-18.

[12] Snapp, S. R., Brentano, J., Dias, G. V., Goan, T. L., Heberlein, T., Ho, C., ... Mansur, D. (1991, October). DIDS (Distributed Intrusion Detection System)–Motivation. In Architecture, and an Early Prototype, InProceedings of the 14th National Computer Security Conference, Washington, DC (pp. 167-176).

[13] Almseidin, M., Alzubi, M., Kovacs, S., Alkasassbeh, M. (2017, September). Evaluation of machine learning algorithms for intrusion detection system. In 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY) (pp. 000277-000282). IEEE.

[14] Ashoor, A. S., Gore, S. (2011). Importance of intrusion detection system (IDS). International Journal of Scientific and Engineering Research, 2(1), 1-4.

[15] Onat, I., Miri, A. (2005, August). An intrusion detection system for wireless sensor networks. In WiMob'2005), IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005. (Vol. 3, pp. 253-259). IEEE.

[16] Zanero, S., Savaresi, S. M. (2004, March). Unsupervised learning techniques for an intrusion detection system. In Proceedings of the 2004 ACM symposium on Applied computing (pp. 412-419).

[17] Altwaijry, H. (2013). Bayesian based intrusion detection system. In IAENG transactions on engineering technologies (pp. 29-44). Springer, Dordrecht.

[18] Dhanabal, L., Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. International journal of advanced research in computer and communication engineering, 4(6), 446-452.

[19] Farnaaz, N., Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213-217.

[20] Huang, Y. A., Lee, W. (2003, October). A cooperative intrusion detection system for ad hoc networks. In Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks (pp. 135-147).

[21] Kaja, N., Shaout, A., Ma, D. (2019). An intelligent intrusion detection system. Applied Intelligence, 49(9), 3235-3247.

[22] Aydin, M. A., Zaim, A. H., Ceylan, K. G. (2009). A hybrid intrusion detection system design for computer network security. Computers Electrical Engineering, 35(3), 517-526.

[23] Smys, S., Basar, A., Wang, H. (2020). Hybrid intrusion detection system for internet of things (IoT). Journal of ISMAC, 2(04), 190-199.

[24] Debar, H., Becker, M., Siboni, D. (1992, May). A neural network component for an intrusion detection system. In Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy (pp. 240-240). IEEE Computer Society.

[25] Depren, O., Topallar, M., Anarim, E., Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert systems with Applications, 29(4), 713-722.