Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

SICAEIT
2024

Research Article
Collection: Artificial Intelligence and Emerging Technologies

Securing the Road: Advancing Cybersecurity in Internet of Vehicles with Deep Learning

Nida Aslam[1], Rizwan Ali Shah[1], Syed Ali Nawaz[1*], and Mubasher H. Malik[2]

[1]Department of Information Technology, The Islamia University of Bahawalpur (IUB), Bahawalpur 63100, Pakistan.
[2]Department of Computer Science, Institute of Southern Punjab, Multan, Pakistan.
*Corresponding Author: Syed Ali Nawaz. Email: ali.nawaz@iub.edu.pk

Academic Editor: Salman Qadri **Published:** February 01, 2024

Abstract: The Internet of Automobiles (IoA) facilitates the exchange of safety-related messages among vehicles, thereby reducing road accidents. Nevertheless, this communication network is susceptible to a number of threats, including erroneous alerts and mispositioning of the vehicle. This paper addresses challenge of authenticating messages to distinguish normal packets from attack packets by proposing an approach to deep learning with binary classification. We utilize Rectified Linear Unit (ReLU) activation algorithms in conjunction with SoftMax classifiers for structured deep neural network to classify normal and malicious packets. The training dataset, prepared from KDD99 and CICIDS 2018 datasets, comprises 120,000 network packets with more than 40 features. Initial preprocessing involves using an autoencoder to eliminate irrelevant data, resulting in 22 valuable features. The Deep Neural Network (DNN) model is trained using Google Colab, utilizing TensorFlow, and validated using a simulated dataset produced via network simulation. Accuracy of investigational findings is 99.48%, that is higher than current models built using convolutional and recurrent neural networks (RNN) and (CNN), respectively. Incorporating sophisticated anomaly detection methods with reinforcement learning approaches may present interesting directions for future study, improving the flexibility and resilience of car communication safety features in ever-changing Internet of Things.The research's primary findings highlight the deep learning-based techniques potential to greatly improve the security and dependability of vehicular communication networks, opening the door for more secure and robust transportation networks in the age of the Internet of Automobiles.

Keywords: Deep Neural Netwrok (DNN); Internet of Automobile (IoA); Deep learning; Binary classification; Cybersecurity.

1. Introduction

The Internet of Automobiles (IoA) serves a critical function in relaying protection messages and upholding the safety of drivers, passengers, pedestrians, and vehicles. Unlike traditional wired networks, which benefit from security measures like gateways and firewalls, wireless networks in vehicles are exposed to safety threats that endanger the overall infrastructure.

As part of the Internet of Vehicles, Vehicular Ad-hoc Networks, or VANETs, operate in an ad hoc fashion and can be the target of many malicious activities, such as impersonation, spamming, message tampering, and communication manipulation. One major challenge is enabling implementation of secure measures inside the Internet of Automobiles (IoA)[1]: careful implementation of security requirements are needed to guard against potential threats from adversaries and malicious automobile networks. An essential part of Auto Net security is intrusion detection, which refers to the capability for recognizing and stopping inappropriate information flows. A number of techniques exist for intrusion detection and prevention, including artificial neural networks, statistical analysis, cluster analysis, and deep learning. Because of its adaptive nature and ability to self-learn, these latter techniques, particularly deep learning, are useful for tracking, identifying and stopping invasions.

It involves communicating with ad hoc vehicle networks, wayside assistance devices, cameras and the vehicle node, as well. Most people agree that detecting intruders safely involves using an intrusion detection system[2]. Every packet that is sent between vehicle nodes must be carefully inspected by the Intrusion Detection System (IDS). This protection system can distinguish between malicious and legitimate activity by using test data from the Internet of Automobiles (IoA).

The basic arrangement of the Internet of Automobiles is depicted in Figure 1 below, where three cars, called OBU1 through OBU3, each have an Intrusion Detection System (IDS) neatly built into their On-Board Unit (OBU) module. These cars can be identified in the graphic as the designated trespasser vehicle; it is the fourth car with the OBU4 label, highlighted in yellow. This system also includes GPS and satellite technology implanted in every automobile to track the exact location of each vehicle. The complete integration of IDS with OBUs ensures continuous monitoring for any attempts at unauthorized access or intrusions within the cars, hence fortifying the security and safety measures incorporated into the Internet of Automobiles architecture.

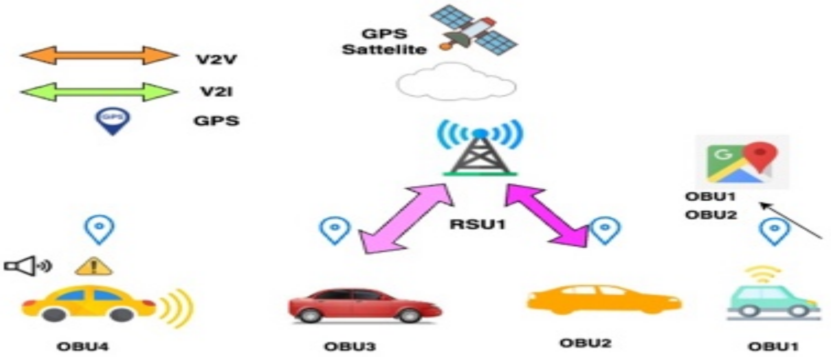

Figure 1. Internet of Automobile architecture showing various components for improved security.

Figure 1 above demonstrates the standard Internet of Automobiles architecture, featuring automobiles equipped with the Intrusion Detection System which is easily integrated into the On-Board Unit (OBU) module. Moreover, Simulation of Urban Mobility simulator, facilitated creation of movement on networks and infrastructure on roads[3]. Diverse mockup scenarios encompassing networks with moderate, low, and significant high densities were executed, encompassing variations in network parameters such as network size, packet size, transmission limit, routing techniques, etc.

Within the field of artificial intelligence, deep learning is a subset of machine learning that has a big impact[4]. Input, output, and hidden layers are the three stages of a neural network's structure that make up the framework of deep learning. By using large datasets for learning, this structure improves efficiency in operation and outperforms traditional machine learning techniques. Additionally, deep learning demonstrates independent classification of features across datasets, enabling further development. Neural network layers contributing to computation and communication in DL include optimization, pooling, convolution, and SoftMax layers. The way these levels interact and are arranged is crucial, and in order to keep the network stable, weight optimization is required after every epoch, or repetition.

Differences in faults are assessed at the analysis stage, and following every round, the matching weights are modified. Through the use of this flexible technique, networks can acquire knowledge more effectively, leading to improved accuracy and precision. The processing capacity required by Deep Neural Networks (DNN) is far more than that of traditional CPUs. As a result, a dedicated Graphical Processing Unit (GPU) is utilized. To develop and implement machine learning and deep learning applications, Google provides Google Collaboratory. Massive datasets may be accessed quickly and easily from remote servers using Google Cloud Storage thanks to this system, which supports several Python versions and execution environments. Google Collaboratory offers a high-performance computing environment for testing and preserving learning models after the necessary files have been mounted and obtained.

Network intrusion detection is a critical protection in automotive networks that safeguards onboard and wayside equipment. Recently, Deep Neural Networks—a branch of deep learning—have received a lot of attention in network intrusion detection research. Numerous research has demonstrated the efficient use of deep learning methods to address challenges in network detection of intrusions. In addition to input and output layers, widely used deep neural networks (DNNs) incorporate multiple hidden layers.[5]. With improved data from a DNN model, the KDD-CUP 99 dataset showed satisfactory performance, with a 99.1% probability of detection and a 0.08 false alarm rate The successful network intrusion detection by the researchers [6] was achieved using accelerated-DNN model, with supervised learning through autoencoders and SoftMax layers. Scientists in the survey [7] of this investigation used improved random forests and support vector (SVM) models on the NSL-KDD dataset to achieve 96% detection probability with a 4% false detection probability.

Investigators demonstrated a recursive neural network-based network surveillance system with a 3.4% false alarm rate and 72.95% accuracy rate on the NSLKDD dataset. The investigators of this work also raised a combination approach based on LSTM-RNN that achieved 90% detection with a 16% false alarm rate on the ADFA dataset. Also, deep belief networks (DBN) have been used to detect intrusions in network systems [8].

Using random forests to pick characteristics based on fluctuating significance rankings, the investigators [9],123] have developed novel machine learning methods for attack prediction. Implementation of support vector machines with the chosen features was evaluated by the investigators [33] on KDD 99 data set, who demonstrate its effectiveness in comparison to classifiers using entire feature sets.

In this research our contribution is as follows:

Proposed novel deep learnig based algorithm for the detection of intrusion.

Introduced advanced autoencoder based data preprocessing technique to enhance security and efficiency for our detection model.

Proposed deep learning model achieved a high rate of accuracy as compared to existing neural network models.

2. Materials and Methods

Effective detection of actions and behaviors in vehicular networks is facilitated through network intrusion detection. Google Colab, provided by Google, is accessible to consumers with a Google Mail account. It provides functionality for both Tensor Processing Units (TPU) and Graphical Processing Units (GPU).

Using captured network traffic, KDD'99[9]creates a dataset with 40 attributes for each network connection. Our research utilizes the KDD'99 dataset to examine how well the suggested classifier separates legitimate packets from malicious packets for intrusion detection. The dataset It includes 23 different output classes, with one class representing normal network connections and the remaining 22 classes representing numerous kinds of malicious networks. Distribution of records includes 97,277 (19.69%) normal links, 391,458 (79.24%) Denial-of-Service (DOS) attacks, and 4,107 (0.83%).

Published in the year 2018 the CICIDS2018 dataset is a Network Intrusion Detection System (NIDS) from the Canadian Institute for Cybersecurity[10]. Upon request, it can be accessed publicly on the internet and includes both current and benign common assaults that mimic real-world data. From network traffic monitoring, the dataset's labelled flows are based on timestamps, source and destination IP addresses, source and destination ports, protocols, and threat categories. The dataset consists of numerous samples evenly distributed between normal traffic (8,000 samples) and various attacks such as Denial-of-Service (DoS)etc. Both of these datasets have been used for training and testing our Deep Neural Network model. The total amount of datasets for training, testing, and validation across CICIDS, KDD 99, and simulation is summarized in Table 1 below. Based on packet types, the dataset is divided into two categories: No Intervention and Intervention. The distribution of data for each type of packet in the training, testing, and validation sets is clearly shown by the aggregate findings.

Table 1. Showcasing training, testing, and validation datasets for CICIDS, KDD 99, and Simulation, categorized by packet type

Dataset	**CICIDS**		**KDD 99**		**Simulation**
Packet type	Training set	Testing set	Training set	Testing set	Validation set
No Intervention	20300	5000	22279	5000	9500

Intervention	20490	4100	24912	4100	7000
Aggregate Result	50106	9000	47191	9000	15000

A brief summary of the training, testing, and validation datasets from KDD 99, Simulation, and CICIDS is given in Table 1 above, which is divided into categories based on the kind of packet. The Network Simulator simulates transport using an On-Board Unit and Roadside Unit in a range of dense or large systems, from lower-density to higher-density networks, with 40 to 500 automobile nodes and 20 to 40 trespasser connections. Various assaults are used to produce trespassers, which are then reported in the log file alongside regular packets. Packets taken from the other two resources contain some incorporated characteristics from the simulation. To standardize the datasets, the log information files have been converted to a single data format. The effectiveness of the deep learning model is evaluated using these simulated datasets, especially for intruder detection in self-driving or self-driving automobiles[11]

2.1. Proposed Methodology

Our suggested preventive method for automotive networks is built on Deep Neural Networks (DNN). The method of detecting intrusions is split into different steps: the stage of preprocessing, stage of feature extraction, and the classification stage. The KDD dataset, which has more than 40 attributes for every network data packet, is the dataset used in the present research. One important step is to identify characteristics that have a high potential for intrusion research. Notable features are regarded as important contributions to this study since they influence the discovery process and reduce the percentage of expected features obtained from the databases. The number of characteristics chosen has a significant impact on execution accuracy and speed.

Preparation includes filtering and data normalization. Every packet number is normalized to fall between 0.0 and 1.0 using equation (1). This normalization of numerical data is assured to yield the best predictor for structured data Deep Neural Network (DNN) training.

to calculate the z-score, which expresses the standard deviation from an observation's departure from the mean. The z-score is calculated using the equation that follows:

$$\boldsymbol{z} = \frac{\boldsymbol{x-u}}{\acute{o}} \tag{1}$$

Here $\boldsymbol{z}$ is the dataset's $\boldsymbol{z-score}$, which runs between 0 to 1, and $\boldsymbol{x}$ is the observation's true value. The standard deviation is represented by ó, and the mean of all values is represented by µ. The lower and upper boundaries of the sigmoid activation function are fitted by these values. The primary characteristics are chosen with the goals of increasing detection classifier accuracy and precision and lowering false alarm rates. Using the idea of Optimization Selection (POS) approach, mathematical evaluation is utilized to choose primary attributes alongside elevated weight and substantial influence during feature selection stage. On the other hand, the Intrusion Detection System (IDS) becomes more efficient overall when superfluous characteristics are removed since they increase the detection rate, decrease computation time, and demands less memory. The needed time drops by 10% and the memory consumption drops by 25% with the inclusion of 15 attributes. The sample of code below shows how to create a neural network, add 11 nodes to the primary layer, and specify the form of the input characteristics. Figure 2 shows how each node functions or makes sense.

$\boldsymbol{Model} = \boldsymbol{Sequential}()$

$\boldsymbol{Model} = \boldsymbol{add}(\boldsymbol{Dense}(\mathbf{10}, \boldsymbol{input\_dim} = \boldsymbol{x.shape}[\mathbf{1}], \boldsymbol{activation} = '\boldsymbol{relu}'))$

$\boldsymbol{Sequential}$ () Initializes a neural network model where layers are added sequentially.$\boldsymbol{Dense}$ $(\mathbf{10})$ Adds a fully connected layer with 10 neurons. $\boldsymbol{input\_dim} = \boldsymbol{x.shape}[\mathbf{1}]$ specifies the input dimension of the layer, which is the number of features in the input data 'x' $\boldsymbol{activation} = '\boldsymbol{relu}'$Sets the activation function of the layer to Rectified Linear Unit (ReLU), which introduces non-linearity to the model by outputting the input directly if it is positive, otherwise outputting zero.

During the classification phase, the self-learning module's acquired features are used to determine the final test results. The classification module uses a SoftMax classifier, whose construction is carried out with the help of the subsequent statement.

$\boldsymbol{Model} = \boldsymbol{add}(\boldsymbol{Dense}(\boldsymbol{y.shape}, \boldsymbol{activation} = '\boldsymbol{softmax}'))$

Out of the more than 40 features available in dataset, required features that have a high potential to identify trespassers are selected. These features include symbolic quantities (e.g., protocol), which need to be changed by conveying each exclusive function a specific number. The node logic shown in Figure 2 down below, includes the calculation $H1j = \sum i = 1n(xi \cdot w1i) + b1$, where $H1j$ denotes the node's output. The weighted sum of the input values, $x\ i$ multiplied by their corresponding weights, $w\ 1\ i$, plus a bias factor, b_1, are the components of this calculation. The resultant value is then subjected to the activation function $H\ 11\ =\ 1\ +\ e\ -H\ 1j$, where $H\ 11\ =\ 1 + e\ \ 1$ and using the sigmoid function to limit the result to a range between 0 and 1. Neural networks frequently use this kind of node logic to process input data and produce an output signal that may be analysed or interpreted further.

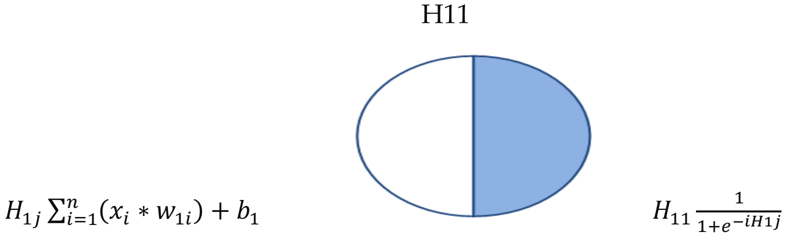
H11
$H_{1j}\sum_{i=1}^{n}(x_i * w_{1i}) + b_1$
$H_{11}\frac{1}{1+e^{-iH1j}}$

Figure 2. Neural Network node computation and activation

Figure 2, shown above, illustrates the computational process and activation function utilized, within a node in a neural network. Ten percent of the 493,300-link training collection was used for the dataset, we used in this investigation. All of the links in the labelled sequence, or about 4 million connections, make up our test set. This lets us use the whole dataset to test the program on unforeseen links. Six states are used to develop rules that properly categorize six separate assault categories up to the present execution stage. The two attack groups and 10% training dataset, yielded the following top three label distributions for attacks: Satan, Ipsweep, and Portsweep.

$$\boldsymbol{k = 2}$$

$$\boldsymbol{Error\ tot} = \sum_{2}^{1} \frac{\boldsymbol{(target - output)2}}{\boldsymbol{k = 1}} \qquad (2)$$

$\boldsymbol{Error\ tot}$ is calculated, along with the target label values and output value.

$$\boldsymbol{Wnew\ =\ Wold\ -\ (\eta\ *\ Errortot)} \qquad (3)$$

Here $\boldsymbol{\eta}$ *is learning rate*. The reliability of the model, diagnosing behaviors, and learning pace are all impacted.

The use of the KDD and CICIDS datasets for feature extraction to train and test a Deep Neural Network (DNN) is demonstrated in Figure 3 below, which demonstrates an enhanced method of intrusion detection in vehicular networks. Furthermore, network simulator capabilities are used to improve the DNN's functionality and guarantee the creation of strong intrusion prevention system.

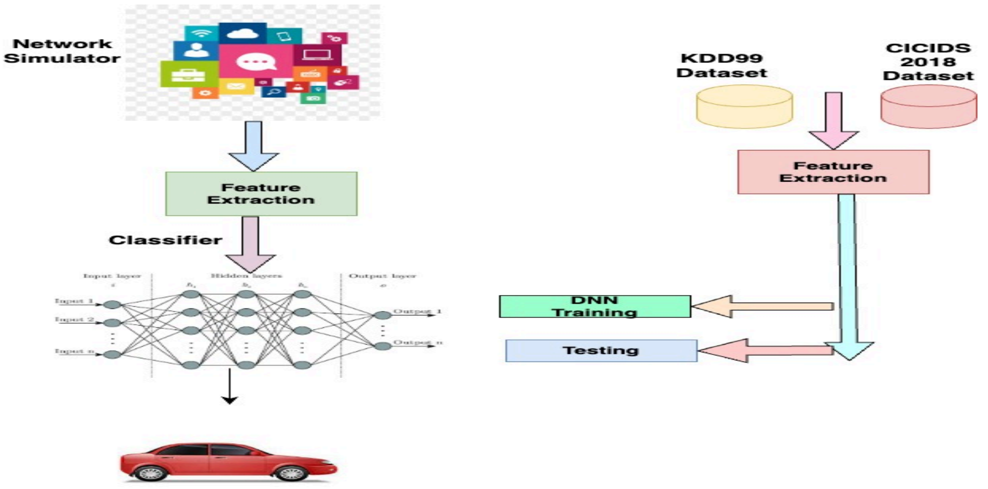

Figure 3. DNN-based intrusion detection in vehicle system.

Figure 3, above shows strengthening DNN-based intrusion detection in vehicle networks by utilizing datasets and network simulator features. In the training phase, 80% of each dataset was utilized, while the remaining 20% was dedicated to testing the model. Furthermore, the suggested model was validated using the data set created by the network simulation. The accuracy of a normal or attack classifier model is determined by classifying all of the analyzed packets using that model. Equation below illustrates the accuracy assessment and computation[12].

$$\boldsymbol{Accuracy\ =\ P\ (Correctly\ Identified\ as\ Attack +\ P\ (Correctly\ Identified\ as\ Normal)}$$
$$\boldsymbol{Sensitivity\ =\ (True\ Positive\ Rate): P\ (Correctly\ Identified\ as\ Attack)}$$
$$\boldsymbol{Specificity\ =\ (True\ Negative\ Rate): P\ (Correctly\ Identified\ as\ Normal)}$$

It is insufficient to gauge the model's performance only by its accuracy. Confusion matrix provides numerous functioning metrics, such as precision and recall, in addition to the F1-Score, which provides a balanced evaluation by considering equivalently significant to both actual and predicted true and false data. The confusion matrix created to solve the shortcomings of accuracy measurement is displayed in Table 2 below. Four important indications are shown in this matrix, which tabulates the results of the model predictions: False Attack (FA), False Normal (FN), True Attack (TA), and True Normal (TN). These numbers are important corroboration indicators that provide a more nuanced evaluation of accuracy by indicating how the model's predictions vary depending on the circumstances.

Table 2. A Confusion Matrix illustrating the outcomes of model predictions.

Datasets		**Model Prediction**	
		1	**0**
Values in matri	Attack values	True Attack value (TA)	False Normal value (FN)
	Normal value	False Attack value (FA)	True Normal value (TN)

Table 2 above shows the four possible outcomes are false attack, false normal, true attack, and false normal that deviate from the model predictions and are utilized as various corroborating indicators to assess accuracy[13] .Equation (4) and (5) are used in accuracy tests to differentiate between the identification of attack and normal packets.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{Total\ correct\ predictions}}{\boldsymbol{Total\ items\ participated}} = \frac{\boldsymbol{(TA+TN)}}{\mathbf{(TA+TN+FA+FN)}} \tag{4}$$

Equation (4) is used to determine the model's accuracy. It calculates the percentage of each test batch sample that can be accurately recognized as threat samples. It expresses, in particular, the proportion of accurately anticipated attack packets to all predicted attack packets.

$$\boldsymbol{Precision} = \frac{\boldsymbol{Actual\ predicted\ Attack}}{\boldsymbol{Total\ attack\ predictions}} = \frac{\boldsymbol{TA}}{\boldsymbol{TA+FA}} \tag{5}$$

Recall is one performance indicator of the suggested approach that is evaluated using equation (6). Recall, which shows the percentage of accurately labeled assault, a reliability metric for the classifier is the number of instances of each assault sample in the test set. It measures the percentage of real attacked packets that are accurately identified as targeted; it is also known as sensitivity.

$$\boldsymbol{Recall} = \frac{\boldsymbol{Predicted\ attack}}{\boldsymbol{Total\ actual\ attacks}} = \frac{\boldsymbol{TA}}{\boldsymbol{TA+FN}} \tag{6}$$

Equation (7) is used to get the recall and precision scores' harmonic mean, or the F1 score. To put it simply, the F1 score offers a thorough evaluation of the model's accuracy by weighing true and false predictions to help identify the optimal model.

$$\boldsymbol{F1\ Score} = \frac{\boldsymbol{2X\ Precision\ X\ Recall}}{\boldsymbol{Precision+Recall}} \tag{7}$$

3. Results

Figure 4 compares the findings of multiple methods used to assess the Intrusion Detection System model's overall detection capability for intrusion data. Determining the assessment metrics of the Deep Neural Network (DNN)-based network intrusion detection model accurately is critical to its efficacy. Greater values of accuracy, precision, recall, and F1-score indicate a lesser False Alarm Rate, reflecting efficiency of model. In an ideal classification scenario, precision and recall would be 1, and the False Alarm Rate would be 0.

The dataset comprises 120,000 data points and 40 plus characteristics. Amid 22 diverse output classes, the non-intrusive class represents legitimate contact links[14], remaining 20 classes depict numerous kinds of malicious links. Most of the data points belong to "normal" set called good connections, accounting for approximately 62%. The "Neptune" class (35.594%) and "back" class are prevalent among the categories representing bad connections (0.650%). Also, some classes have minimal data points, each with lesser than 10 instances per class, having lowermost amount. Given that unequal distribution of data points, it is essential to develop a model capable of accurately classifying all those points that consist of these diverse groups.

Figure 5 presents a comparative evaluation [15] of suggested model that is based on Deep Neural Networks (DNN) regarding F1-score, recall, accuracy, and precision. With a precision score of 99.48% Figure 4 below illustrates the better performance of our Deep Neural Network (DNN) model that was trained on the KDD 99 dataset. This is superior to both the accuracy reported in the literature at 78.24% and the Convolutional Neural Network (CNN) model, which achieved 96.62% precision.

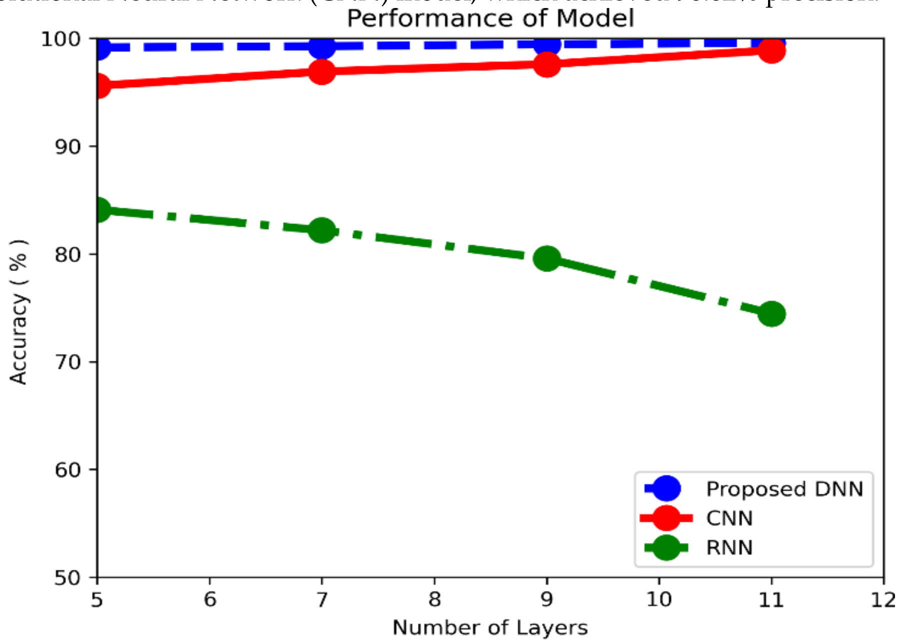

Figure 4. Deep Neural Network (DNN) Model Performance Comparison with various algorithms.

Figure 4 above shows a deep neural network model performance comparison with various algorithms. A performance comparison of models, including our suggested Deep Neural Network (DNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN), is displayed in Figure 5 below. Evaluation metrics are employed to evaluate the effectiveness of each model in terms of intrusion detection inside vehicular networks. These metrics include precision, recall, and F1-score. Our DNN model performs better on a variety of criteria, proving that it is useful for improving security protocols and guaranteeing strong intrusion detection.

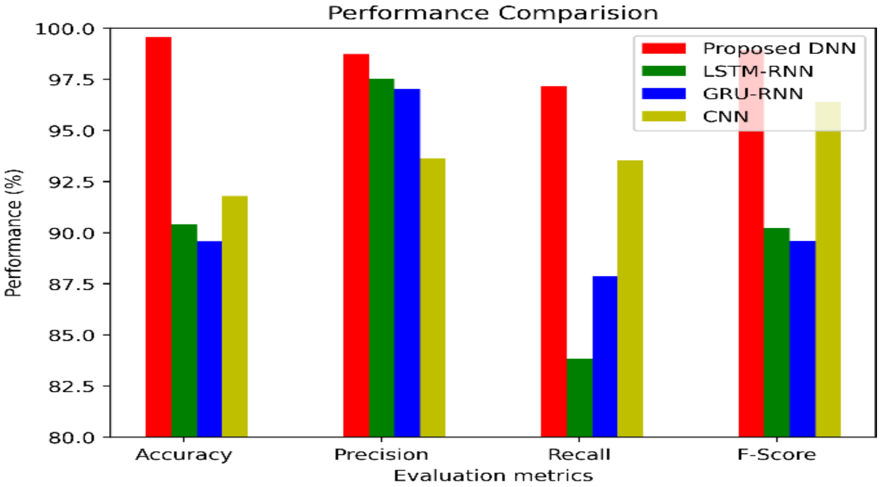

Figure 5. Contrast of evaluation metrics in various models.

Figure 5 above shows evaluation matrices like accuracy, F1 score, and recall comparison with various models. The accuracy metrics attained by several algorithms, such as the Grated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and our suggested Deep Neural Network (DNN), are compared in the Table 3 below. Notably, with an astounding accuracy score of

99.48%, the suggested DNN model beats all other methods. This demonstrates how well our method works to provide better intrusion detection performance in automotive networks.

Table 3. Accuracy comparison of algorithms, highlighting proposed DNN in network intrusion detection.

Sr.	Algorithm	Accuracy
1	Grated Recurrent Unit (GRU)	88.4%
2	Long Short-Term Memory (LSTM)	91.7%
3	Convolutional Neural Network (CNN)	96.2%
4	Proposed Deep Neural Network (DNN)	99.48%

The table 3 above compares accuracy metrics of diverse algorithms, highlighting the proposed Deep Neural Network (DNN). The DNN outperforms the other techniques in the table with an accuracy of 99.48%. In order to demonstrate how well different algorithms perform in accurately recognizing positive examples, seizing all positive occurrences, and maintaining an overall balance between precision and recall, Metrics for recall, precision, and F1 score are shown for each approach utilized in the classification job in Table 4 below. Among the proposed algorithms, the Deep Neural Network (DNN) performs the best overall, with an F1 score of 99%, recall of 97%, and precision of 98%, clearly outperforming the others in the evaluation criteria.

Table 4. Evaluation matrices comparison with proposed model

Sr	Recall	Precision	F1 Score
GRU-RNN	82%	87%	87%
LSTM-RNN	83%	85%	85%
CNN	92%	94%	96%
Proposed DNN	97%	98%	99%

Metrics for recall, precision and F1 score are shown for each approach utilized in the classification job in Table 4 above.

4. Conclusion and Future Work

The proposal and assessment of a novel intrusion prevention technique created especially for the Internet of Automobiles (IoA) are discussed in this paper. Our technology is designed to handle the particular security difficulties presented by networked automobile systems, guaranteeing the secure and dependable transfer of critical information within automobile networks.

The results of our test represent the robustness and effectiveness of our intrusion prevention system on a variety of operating settings, corroborating our rigorous assessment process following extensive hands-on with military datasets. Not only do our results show an incredible accuracy of 99.48%, this stage of precision demonstrates how well the suggested model is at identifying and mitigating potential vulnerabilities in IoT systems.

Our research roadmap provides a range of potential future paths for enhancing and refining our safeguarding system. We are committed to exploring fresh datasets in order to increase the model's training material and improve its adaptability in real-time. Furthermore, we want to tailor the model to optimize feature selection and reduce model training duration, hence enhancing net performance. To find ways to enhance our system's effectiveness across a range of IoA instances, we are also eager to explore additional advanced machine learning techniques including ResNet, EfficientNet, and Deep Belief Networks (DBNs). In conclusion, our technique for preventing intrusions represents a significant advancement in safeguarding the reliability and safety of car networks connected to the Internet of Cars.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. G. Liu and J. Zhang, "CNID: Research of Network Intrusion Detection Based on Convolutional Neural Network," Discrete Dyn Nat Soc, vol. 2020, 2020, doi: 10.1155/2020/4705982.
2. M. El Boujnouni and M. Jedra, "New intrusion detection system based on support vector domain description with information gain metric," International Journal of Network Security, vol. 20, no. 1, pp. 25–34, Jan. 2018, doi: 10.6633/IJNS.201801.20(1).04.
3. P. A. Lopez et al., "Microscopic Traffic Simulation using SUMO," International Conference on Intelligent Transportation Systems, vol. 2018-November, pp. 2575–2582, Dec. 2018, doi: 10.1109/ITSC.2018.8569938.
4. "Deep Learning Specialization - DeepLearning.AI." Accessed: Feb. 16, 2024. [Online]. Available: https://www.deeplearning.ai/courses/deep-learning-specialization/
5. W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," Neural Comput, vol. 29, no. 9, pp. 2352–2449, Sep. 2017, doi: 10.1162/NECO_A_00990.
6. D. H. Shin, K. K. An, S. C. Choi, and H.-K. Choi, "Malicious Traffic Detection Using K-means," The Journal of Korean Institute of Communications and Information Sciences, vol. 41, no. 2, pp. 277–284, Feb. 2016, doi: 10.7840/KICS.2016.41.2.277.
7. Y. Chang, W. Li, and Z. Yang, "Network Intrusion Detection Based on Random Forest and Support Vector Machine," 22017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol. 1, pp. 635–638, Aug. 2017, doi: 10.1109/CSE-EUC.2017.118.
8. W. Elmasry, A. Akbulut, and A. H. Zaim, "Empirical study on multiclass classification-based network intrusion detection," Comput Intell, vol. 35, no. 4, pp. 919–954, Nov. 2019, doi: 10.1111/COIN.12220.
9. "KDD Cup 1999 Data." Accessed: Feb. 16, 2024. [Online]. Available: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
10. "IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB." Accessed: Feb. 16, 2024. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html
11. K. M. Ali Alheeti and K. McDonald-Maier, "Hybrid intrusion detection in connected self-driving vehicles," 2016 22nd International Conference on Automation and Computing, ICAC 2016: Tackling the New Challenges in Automation and Computing, pp. 456–461, Oct. 2016, doi: 10.1109/ICONAC.2016.7604962.
12. B. Cao, C. Li, Y. Song, Y. Qin, and C. Chen, "Network Intrusion Detection Model Based on CNN and GRU," Applied Sciences (Switzerland), vol. 12, no. 9, May 2022, doi: 10.3390/APP12094184.
13. Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier," Computer Networks, vol. 174, Apr. 2019, doi: 10.1016/j.comnet.2020.107247.
14. J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated Feedback Recurrent Neural Networks," Feb. 2015, Accessed: Feb. 16, 2024. [Online]. Available: http://arxiv.org/abs/1502.02367
15. B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," Proceedings of 2016 8th IEEE International Conference on Communication Software and Networks, ICCSN 2016, pp. 581–585, Oct. 2016, doi: 10.1109/ICCSN.2016.7586590.