Innovative: International Multi-disciplinary Journal of Applied Technology
(ISSN 2995-486X) VOLUME 02 ISSUE 07, 2024

Federated Learning-Based Architecture for Detecting Cyber Vehicular Attackers in Intelligent Transportation Systems

Dalia Abdulrahim Mokheef Aljabri
University of Babylon , College of Basic Education, Department of Mathematics

Maki Mahdi Abdulhasan
Al-Nisour University College, Baghdad, Iraq

Noor R. Obeid
University of Babylon, College of Information Technology, Department of Information Security

Abstract:

The Internet of Vehicles (IoV) concept within Intelligent Transportation Systems (ITS) integrates automobiles, transportation, information sharing, and traffic infrastructure management to enhance road safety. IOV can collaborate to create models through federated learning, improving performance, enhancing data privacy, and ensuring the security of local vehicle data. This paper introduces a novel Distillation-based Semi supervised (DS-FL) Model for intrusion detection. This model was demonstrated using the datasets to address the heterogeneity and diversity of devices and malicious samples. Experimental results show that the proposed system achieved 99.48%, 99.75%, 99.83%, and 99.93% accuracy in detecting various types of attacks on the ISCXIDS2012, CIC-IDS2017, CSE-CIC-IDS2018, and Car-Hacking datasets, respectively, outperforming other intrusion detection techniques. It highlights the model's effectiveness in securing intelligent transportation system networks against cyber-attacks.

Keywords: *Data Analytics, Federated Learning, ITS, IOV, Vehicular Networks.*

1. Introduction

The rapid expansion of the internet of Things (IoT) has enabled novel requests, including smart grids, smart cities, and IoV[1]. When consistent vehicles operate over cyberspace, IoT evolves into

IoV. Important advancements in smart automobile technology have brought considerable attention to IoV technologies, further driving innovation in this field. A vehicle-to-vehicle (V2V) network is a combined and open system that attaches vehicles with human intelligence, close atmospheres, and community networks. Human error and congestion are reduced by these networks. Despite the IoV's many benefits, several issues must be considered to ensure the safety of all thoroughfare users [2].

The IoV is susceptible to cyberattacks, which can disrupt its constancy and heftiness and result in vehicle unavailability and traffic accidents. Multi-component networks require multiple components for communication, so they are helpless to various attacks. A comprehensive intrusion detection system (IDS) can detect probable cyberattacks and protect the network from various types of attacks. A network intrusion detection system identifies differences and spells in data during vehicle-to-device communications[3]. The IoV is a moderately new network example, so it faces continuously budding attacks. The IoV network generates data rapidly, especially during cyberattacks. In this high-stakes environment, appliance learning and deep learning approaches are favorite [4], [5].

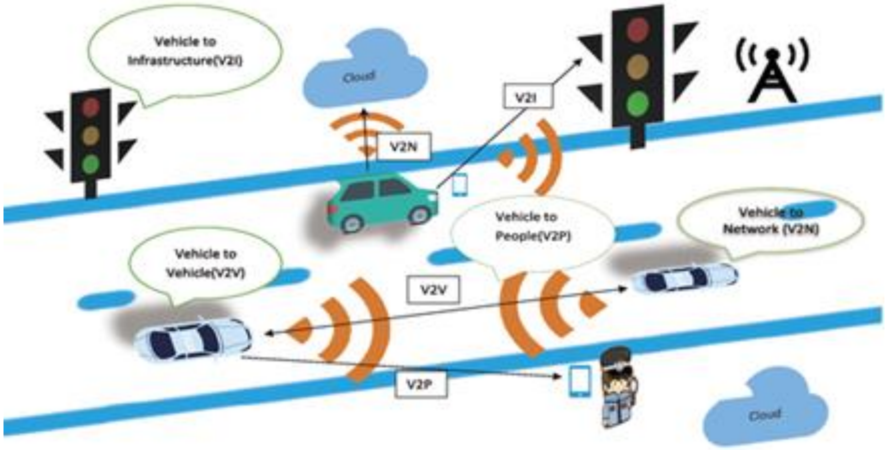

Figure 1: The communications model for the Internet of vehicles

As the IoV moves toward a fully connected era, road accidents are expected to be reduced, overall transport safety will improve, transportation comfort will be enhanced, traffic congestion will be relieved, and environmental impact will be significantly reduced. Bright vehicles in the IoV are expected to be prepared with over 100 multisensory to facilitate efficient communications [6], [7]. [6], [8] such as V2V, Vehicle-to-Substructure (V2I), etc. are presented in Figure 1. 5G mobile networks promise advanced features to enhance effective communication in IoV environments that are characterized by high mobility.

A cooperative machine learning paradigm that prioritizes data privacy and integrity has gained significant significance in recent years: Federated Learning (FL) [9], [10]. In FL, system training is devolved across multiple lumps or customers using their local data. These parameters are exchanged and aggregated to create an individual global model, complete a dominant server or via a peer-to-line [11], [12]. Each client creates an aggregate of all individual models following several iterations.

Deep learning is collaborative and privacy-protected with federated learning, with energy preservation as an additional benefit. Using federated learning to analyze driver behaviour to save training time and maintain data privacy is novel. The capacity and distribution of several data sources are available. An integrated federated learning model was developed and implemented to handle heterogeneous datasets, regardless of whether they are Independent and Identically Distributed (IID). Four system spell uncovering datasets were assessed: the ISC-XIDS2012, CSE-

CICIDS2018, and CIC-110 IDS2017, Car Hacking dataset. A comparison of our attack detection method with other FL-based attack detection methods is conducted to determine its effectiveness.

In the remainder of the paper, the sections are organized as follows. The literature is discussed in section 2. Section 3 presents the proposed methodology. A discussion of the results is presented in Section 4, and the paper concludes in Section 5.

2. LITERATURE REVIEW

Several works have applied FL to IoT security in recent years, with FL gaining importance in cybersecurity. Data privacy problems are evident from the study [13], but the study evaluated private data. It is unlikely that the data will be distributed randomly among clients in realistic scenarios where each client's data will come from a different distribution. In contrast with the previous studies, which focused on network data, the studies in [8] and [9] focused specifically on IoT device applications and sensor understandings [8] and [9]. Author [16] examined FL using intrusion detection systems as a case study. Additionally, blockchain technology is used to mitigate adversarial FL issues. Despite not specifically targeting IoT devices, it focuses on early stages of intrusion detection rather than existing malware detection[17].

Previously, the author developed a C-ITS application server protection system against position falsification attacks using V2N in our previous work [18]. In order to improve discovery results, historical position data is used. A FL and blockchain-based miss behaviour recognition system for VANETs was projected in [19]. An RSU can send a shared exercise model to a vehicle, which can then update the parameters without sharing local data with the RSU. On the blockchain, models are aggregated based on their accuracy scores and stored as an averaged model. According to [20], An extended version of VeReMi was created. Among the 19 attack types included in this expansion are sophisticated attacks like Information Replay, DoS, and Sybil at various vehicle thicknesses. A basic misbehaviour detection mechanism based on LSTMs and DNNs is also used in this study. Further research is based on these results. Researchers can leverage this large dataset to compare and enhance their detection mechanisms and develop new ones.

Machine learning (ML) is an essential component of many AI applications [21]. An ML approach includes RL, SL, and unsupervised learning. The unsupervised machine learning scheme requires untagged data for training. To represent untagged data, it searches for a suitable method. Supervised learning is the other handling based on a set of labelled data. Discrete and continuous data are trained with regression and classification algorithms in supervised learning. Reinforcement learning (RL) studies behaviour from the perspective of constant rewards to take advantage of cumulative rewards. Reinforcement learning is achieved through Markov Decision Processes (MDPs) [14]. Numerous automotive network research problems can be addressed using this plan, such as optimum route prediction for electric vehicles, cooperative optimization of oil consumption within a region, and traffic jam reduction. Several IoV-related contexts are discussed in it [22].

IDSs based on ML may encounter issues that FL can address. An FL approach can enable manifold applicants to develop vigorous and efficient ML copies without sharing data [23]. Federated Learning (FL) is among the most adaptable techniques for training machine learning (ML) algorithms on edge devices. In contrast to other approaches that do not use FL, this strategy preserves user privacy [24]. Algorithms based on FL have several advantages. Using FL, predictive models can be learned, and the training dataset can be maintained instead of centralized [25]. It reduces time consumption and allows access to data without contacting a centralized server. Moreover, its low complexity and distributed architecture make it particularly suitable for deployment on resource-constrained hardware [25].

3. PROPOSED FEDERATED LEARNING-BASED MODEL

In response to these concerns, Federated Learning has taken steps to address them (FL) [9], [26], [27]. A centralized server coordinates the sharing of machine learning models in FL instead of sharing data instances. The coordinator builds a global model that protects confidential information by aggregating local models from each device. Data that contains sensitive information should never be disclosed. While FL offers several benefits over centralized learning, it faces several performance challenges [27]. The limited amount of data generated by each device often leads to performance degradation, which reduces the effectiveness of the device as a whole.

Furthermore, FL devices typically encounter different data types, leading to class imbalances and further affecting performance. When heterogeneous devices participate, performance issues may arise due to observations of different events. A global aggregation server uses edge devices (ED) to produce a learning model [28]. Figure 2 illustrates FL for IoV networks.

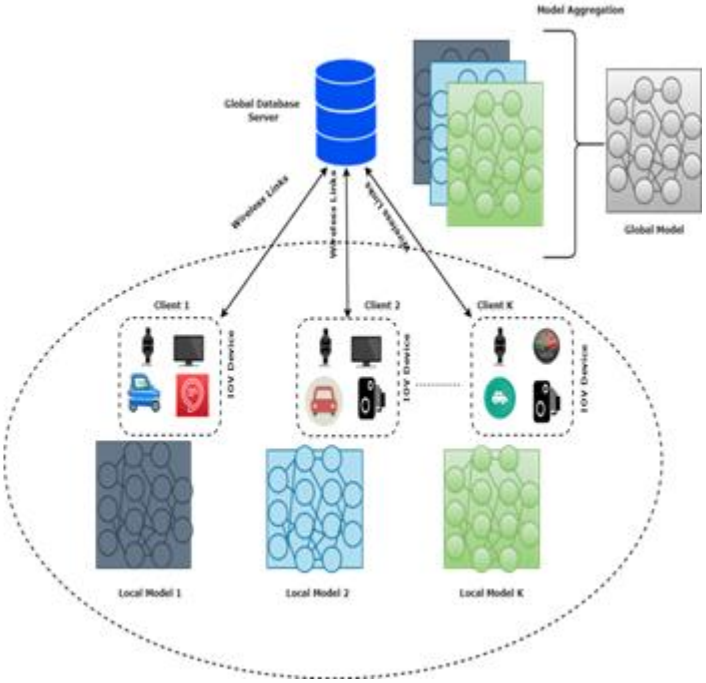

Figure 2: FL model for Vehicular IoV devices.

A. Federated learning Implementation

PyTorch, a research-driven platform, enables using PySyft, an FL-specific framework [29]. Deep learning is made secure and private using PySyft, a Python-based FL implementation. The private data is separated from the model training process by integrating FL and PyTorch machine learning. This setup uses PyTorch as a traditional centralized implementation, while PySyft virtual workers provide a local simulation aggregated at each epoch. The first step is to request a profile from the IoT security service. This simulation will use only the dataset to train the model for testing. In order to train and test virtual workers, we need to create them first. Virtual workers are assigned a location ID before training begins, which assigns the optimizer to their location. The virtual workers have been created, and the dataset has been batched and sent to the working hardware device, or in this case, the simulation of the virtual workers. The security gateway would handle port mirroring to avoid interrupting the signal to the IoT device by using data mirrored over the network-specified port. A virtual worker's benign data is batched evenly during training in the simulation.

The virtual workers begin processing their respective batches of training data once the training data is batch-sending to them. Every worker signals the end of an epoch after completing their tasks. After completing each epoch, a zero gradient is used to clear the optimized gradients. IoT global models are then aggregated from the local models. After the global model has been aggregated, it is sent back to the security gateways for the specific type of device to continue the training process. During training, the output loss obtained from the new global model is aggregated from the local models after each epoch.

B. Distillation-based Semisupervised (DS-FL) Model

The DS-FL model consists of six steps, where each client holds a labelled private dataset Dk and an unlabeled dataset Do=xjo|j=1,2,…,No. Here xjo represents the vectorized feature of a sample, and

No, represent the number of samples in the open dataset. For simplicity, we use a matrix Xoto denote the concatenated vectorized samples of Do.

As shown in Eqn. 1, each client trains its deep learning model with a privately labeled dataset in the first communication round. During the second step, clients K generate predictions for the unlabeled samples, i.e., local logits:

$$\widehat{\boldsymbol{p}}_j^k = F\left(\boldsymbol{x}_j^o|\boldsymbol{w}^k\right) \qquad (5)$$

Here, use the matrix $\widehat{\boldsymbol{P}}^k$ to represent the concatenated $\widehat{\boldsymbol{p}}_j^k| = j = 1,2,\dots,N^o$ of the DL model. A central server then uploads the local logs from each client. Based on these local logits, the server aggregates the global logits $\widehat{\boldsymbol{p}}_j^s$ as follows:

$$\widehat{\boldsymbol{p}}_j^s = \frac{1}{K}\sum_{k=1}^{K}\widehat{\boldsymbol{p}}_j^k \qquad (6)$$

$$\widehat{\boldsymbol{p}}_j^s \leftarrow S\left(\widehat{\boldsymbol{p}}_j^s|T\right) \qquad (7)$$

where $S(\cdot\,|T)$ is the softmax function with the T temperature, i.e.,

$$S\left(\widehat{\boldsymbol{p}}_j^s|T\right) = \frac{exp\left(\frac{\widehat{\boldsymbol{p}}_j^s}{T}\right)}{\sum_{l=0}^{L-1} exp\left(\frac{\widehat{p}_j^s}{T}\right)} \qquad (8)$$

When T is less than 1, the distribution of pj becomes sharper, reducing the information entropy. DS-FL can be accelerated and stabilized in non-IID data distributions by reducing the entropy of global logits [30]. The matrix Ps, which represent the concatenated pjs|=i=1,2,…,No, is then broadcast to each client. Finally, each client trains its DL model with the global logits through the process, called distillation, as follows:

$$\boldsymbol{w}^l \leftarrow \boldsymbol{w}^k - \gamma\nabla\psi\left(\widehat{\boldsymbol{P}}^k, \widehat{\boldsymbol{P}}^s\right) \qquad (9)$$

In addition, the server maintain global model ws and trains it using the shared unlabeled dataset along with the global logits as follows:

$$\boldsymbol{w}^s \leftarrow \boldsymbol{w}^s - \gamma\nabla\psi\left(F(\boldsymbol{X}^o|\boldsymbol{w}^s), \widehat{\boldsymbol{P}}^s\right) \qquad (10)$$

These procedures iterate over multiple communication rounds. Each DS-FL client must train its local model using a private dataset. However, when the trained model predicts on an open dataset, if the distribution of the private dataset is non-IID, the distribution of samples in the open dataset can differ significantly. This mismatch can lead to incorrect predictions by the trained model, reducing the number of local logits generated.

4. Experimental results

Results from several experiments were collected to validate this framework. The initial section of the paper evaluates the performance of supervised and unsupervised methods in detecting malware.

A. Datasets and Pre-Processing

The V2X networks involve communications between the vehicle and other external entities. In-vehicle networks fall under the second category. The datasets used are listed below:

1. ISCXIDS2012 [31]: A heavily skewed class distribution is observed in the dataset, with binary labels "normal" and "attack."
2. CIC-IDS2017 [32]: It includes the HTTPS protocol, which accounts for nearly 70% of network traffic, to improve ISCXIDS2012. In contrast to ISCXIDS2012, this improved dataset has a better distribution of classes and is a multi-class dataset.
3. CSE-CIC-IDS2018 [33]: A category named "Benign" appears in CIC-IDS2017 and CSE-CIC-IDS2018, while the remaining fourteen categories are labelled as attacks.
4. Car-Hacking [34]: Using the vehicle's OBD-II port, malicious traffic is injected into a genuine network in-vehicle, and this dataset pertains to in-vehicle networks. This dataset has four categories of attacks: DOS attacks, fuzzy attacks, spoofing attacks on drive gears, and RPM gauges.

Based on Table 1, the distribution of classes between the datasets is uneven, with 99.20% of samples classified as "normal" and only 2.8% as attack. Similar to CIC-IDS2017, 80.30% of CSE-CIC-IDS2018 samples were classified as "normal," according to CSE-CIC-IDS2018. A total of six categories are used in CIC-IDS2017, and five categories are used in CSE-CIC-IDS2018; 86.6% of samples in the Car Hacking dataset are classified as "normal." In contrast, the remaining samples are distributed among four types of attacks.

Table 1: IOV cyber-attacks described

Dataset	No. of Instances	Attack Class	Label Type
ISCXIDS2012	2,450,324	2.8%	Binary
CIC-IDS2017	2,830,743	19.7%	Multi-class
CSE-CIC-IDS2018	19,233,002	17.0%	Multi-class
CAR-HACKING	17,357,681	13.4%	Multi-class

Hierarchical classification is used in the proposed work, which involves two classification stages. As a first stage, a binary classification determines whether the instance is a "normal" or an "attack". A hierarchical multi-class classification frames the problem in the second stage is classified as an "attack.". Except for the ISCXIDS2012 dataset, which consists only of binary labels and thus only requires the first stage of this process, the labels must be processed in the two stages.

In order to improve network performance, the data range is converted into a fixed scale by normalizing or scaling it. Deep learning requires a matrix format for training data, similar to an image, and normalizing input data using the following equation:

$$d' = \frac{d - \min(d)}{\max(d) - \min(d)}$$

4.1 Oversampling

Oversampling and undersampling were used to resolve imbalances. It is possible to optimize both detection performance and delay with SMOTE filter [35], [36], [37]. Other approach involves linearly combining two comparable minority samples to generate additional data points. By interpolating between a minority sample and its neighbors, new feature values can be calculated.

4.2 Feature Selection

Feature selection systems identify relevant features related to target concepts. Feature extraction and selection enhance and decrease computational complexity learning performance, decrease storage requirements and build more generalizable models. As opposed to feature extraction, feature selection retains and enhances the physical meanings of features by preserving them without transforming them.

Proposed techniques are used in this paper to enhance the reliability and efficiency of the system by selecting prominent features, computing them individually, and averaging the results. Traditional feature selection methods like the Gini coefficient, information gain and entropy are also utilized to assess feature rank through variations in tree-based technique parameters. Features are ranked by reputation, with high-importance features added first until their cumulative importance ranges from 0.9. A computation-intensive feature is excluded if its importance sum is less than 0.1.

B. Performance Metrics

This paper employs three presentation metrics to assess the feature engineering process, prediction accuracy, and sensible presentation. The next quantities were used to evaluate and compare forecast models quantitatively:

The evaluation includes four potential organizations for a sample dataset that classifies usual and spells together. These classifications are True Positives, False Negatives, False Positives, and True Negatives, as illustrated in Table 2. A classification is deemed correct if it is True and incorrect if it is False. Positive classification Positive samples are indicated by this indicator (classifier identifying confident samples), while negative organization denotes the presence of negative samples (classifier identifying Sampling negatives).

1. Right Confident: Correctly detects a usual instance.
2. False Positive: Incorrectly classifies an abnormal instance as normal.
3. An instance classified as abnormal is incorrectly classified as being False Negative.
4. Correctly identifies abnormal instances as true negatives.

Table 2: Detection results

	Relevant	**Not relevant**
Detected	True positives (TP)	False positives (FP)
Not Detected	False Negative (FN)	True negatives (TN)

Accuracy (Acc)**:** It can be calculated as the percentage of instances properly classified as either normal or attacks using the following formula:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1)$$

Precision *P*: A relevant instance can be defined by a percentage among the detected instances. The formula for calculating this is as follows:

$$P = \frac{TP}{TP + FP} \qquad (2)$$

Recollection *R*: Based on the formula below, this figure represents the percentage of relevant instances that have been detected compared to the total amount of relevant instances:

$$R = \frac{TP}{TP + FN} \qquad (3)$$

F1-Score: In order to calculate the F1-Score, consider the following formula:

$$F_1Score = \frac{(\propto^2 + 1)P * R}{\propto^2\ (P + R)} \qquad (4)$$

More specifically, where $\propto = 1,$ the new formula of $F_1\text{Score}$ is

$$F_1Score = \frac{2PR}{P + R} \qquad (5)$$

C. Results Discussion

The proposed algorithm is simulated, and its performance is evaluated using a desktop computer with the following specifications: an Intel® Core™ i7 @ 2.50GHz processor and 16 GB of RAM. An NVIDIA Federated Learning (FL) was trained with GeForce RTX 3080 Ti 16GB graphics cards. Implemented with Keras and Tensor Flow back ends in a Python-based environment, the scheme was simulated in a Python-based environment.

Figure 3 illustrates the results of evaluating the DS-FL model on the CIC-IDS2017, ISCXIDS2012, Car-Hacking, and CSE-CIC-IDS2018 datasets. Among these datasets, the DS-FL model demonstrates superior performance on the Car-Hacking dataset. Specifically, features G, B, F, and H achieved the highest accuracies: 99.4777%, 99.7491%, 99.8244%, and 99.9260%, respectively, using the DS-FL model on ISCXIDS2012, Car-Hacking, CSE-CIC-IDS2018, and CIC-IDS2017 datasets. The DS-FL model shows significantly improved results, particularly on the Car-Hacking dataset. For evaluation purposes, 10 features were randomly selected.

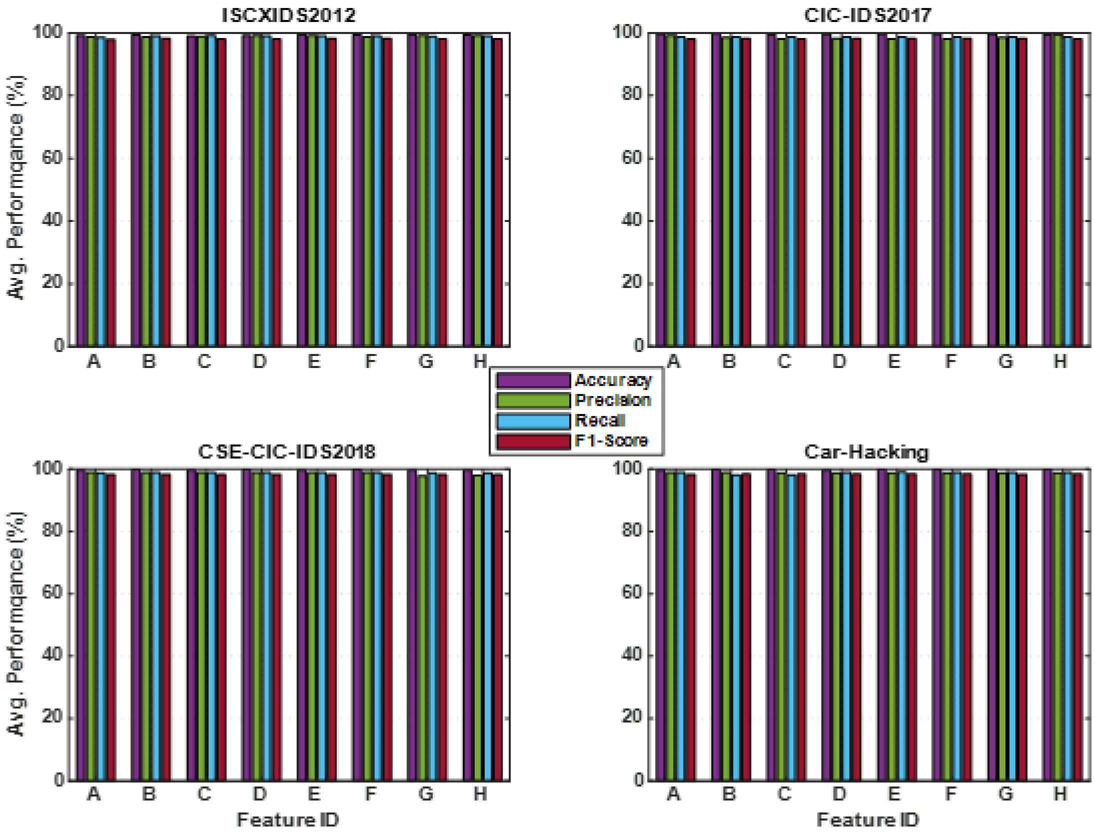

Figure 3: Avg. performance (%) analysis versus Feature ID.

Figure 4 illustrates the accuracy progression as the number of features varies for the ISCXIDS2012, Car-Hacking, CSE-CIC-IDS2018, and CIC-IDS2017 datasets using the DS-FL model proposed in this study. The results demonstrate that the DS-FL model excels, particularly with the Car-Hacking dataset across different feature sets.

For the CIC-IDS2017, CSE-CIC-IDS2018, and Car-Hacking datasets, the DS-FL model achieves a remarkable 99% accuracy. The figure also shows that achieving 100% precision requires approximately 69 features for CIC-IDS2017, 59 for ISCXIDS2012, 50 for Car-Hacking, and 39 for CSE-CIC-IDS2018 datasets. Furthermore, the analysis reveals that around 50 features are adequate to achieve 100% recall and F1-Score for detecting attacks across all datasets.

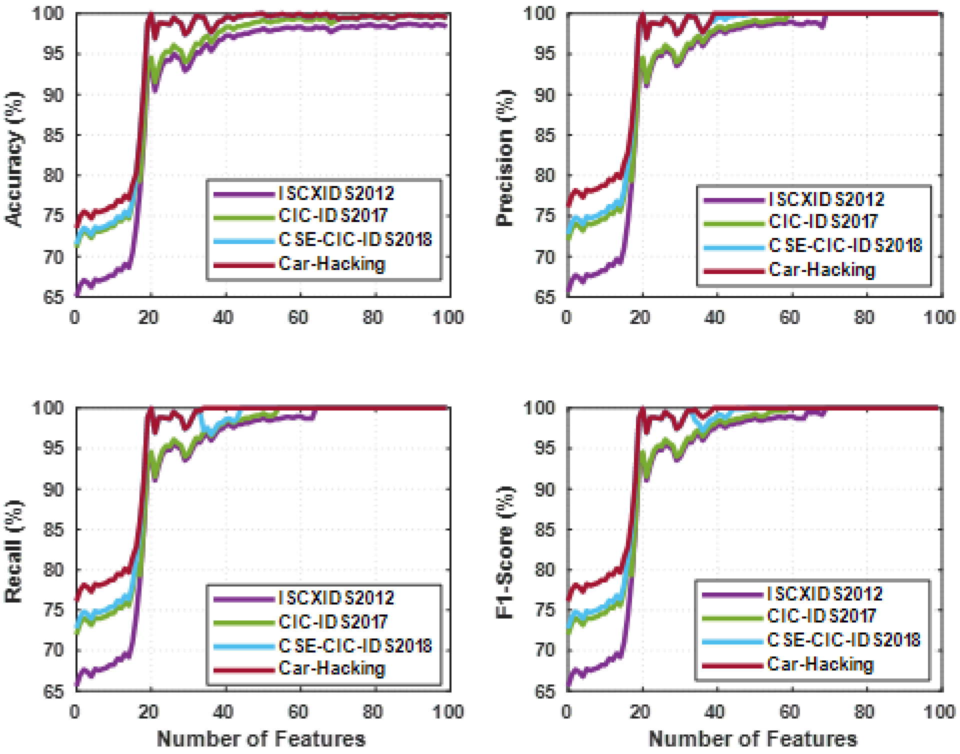

Figure 4: Accuracy, precision, recall and F1-Score (%) versus number of features.

To legalize the performance of the future scheme, we compared its Graphics cards equipped with GeForce RTX 3080 Ti 16GB, which were used for teaching at Federated Learning (FL). [34], LSTM neural-network-based model [38], deep learning-based IDS [39], FED-IDS [40], and A framework for detecting attacks on vehicular sensor networks (VSNs) based on FL [41]. Our proposed model demonstrated superior performance, particularly on the Car Hacking dataset, achieving higher accuracy than the methods listed in Table 3.

Table 3: Compared accuracy (%)to existing algorithms.

Author & Year	Dataset	Accuracy (%)
GIDS [34]	CAN Intrusion	98.0
LSTM-NN[38]	CAN bus	99.11
Deep Learning-based IDS [39]	Car Hacking, UNSW-NB15	99.0
FED-IDS [40]	Car-Hacking, TON_IoT	97.82
FL-VSN[41]	Car Hacking	99.52
Proposed Model	ISCXIDS2012	99.48
	CIC-IDS2017	99.75
	CSE-CIC-IDS2018	99.83
	Car-Hacking	99.93

5. Conclusion

Various cyber-attacks can be carried out against the Internet of Vehicles (IoV) due to their widespread presence and insufficient security measures. Several types of attacks can be prevented with the use of Intrusion Detection Systems (IDSs). A distillation-based semi-supervised (DS-FL) model is used in this study to detect abnormal events in IoV networks and learn the behaviour of normal network traffic. The proposed method leverages statistical features of network behaviour to create a robust learning model of normal traffic flow in IoVs. Evaluation results indicate that As far as accuracy, precision, recall, and F1 scores are concerned, the proposed model performs exceptionally well. The overall normal exactness of the proposed IDS model is 99.48%, 99.75%, 99.83%, and 99.93% for the ISCXIDS2012, Car-Hacking, CSE-CIC-IDS2018, and CIC-IDS2017 datasets, respectively.

REFERENCES

1. P. Rani and R. Sharma, "Intelligent Transportation System Performance Analysis of Indoor and Outdoor Internet of Vehicle (IOV) Applications Towards 5G," *Tsinghua Sci. Technol.*, 2023.
2. P. Rani and R. Sharma, "IMFOCA-IOV: Intelligent Moth Flame Optimization based Clustering Algorithm for Internet of Vehicle," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, 2023, pp. 1–6.
3. P. Rani, N. Hussain, R. A. H. Khan, Y. Sharma, and P. K. Shukla, "Vehicular intelligence system: time-based vehicle next location prediction in software-defined internet of vehicles (SDN-IOV) for the smart cities," *Intell. Things AI-IoT Based Crit.-Appl. Innov.*, pp. 35–54, 2021.
4. T. Alladi, V. Kohli, V. Chamola, and F. R. Yu, "Securing the Internet of Vehicles: A Deep Learning-Based Classification Framework," *IEEE Netw. Lett.*, vol. 3, no. 2, pp. 94–97, Jun. 2021, doi: 10.1109/LNET.2021.3058292.
5. P. Rani and R. Sharma, "An experimental study of IEEE 802.11 n devices for vehicular networks with various propagation loss models," in *International Conference on Signal Processing and Integrated Networks*, Springer, 2022, pp. 125–135.
6. J. Eze, S. Zhang, E. Liu, and E. Eze, "Cognitive radio technology assisted vehicular ad-hoc networks (VANETs): Current status, challenges, and research trends," in *2017 23rd International Conference on Automation and Computing (ICAC)*, Huddersfield, United Kingdom: IEEE, Sep. 2017, pp. 1–6. doi: 10.23919/IConAC.2017.8082035.
7. R. Muraleedharan and L. A. Osadciw, "Cognitive security protocol for sensor based VANET using swarm intelligence," in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA: IEEE, 2009, pp. 288–290. doi: 10.1109/ACSSC.2009.5470101.
8. M. Harounabadi, D. M. Soleymani, S. Bhadauria, M. Leyh, and E. Roth-Mandutz, "V2X in 3GPP Standardization: NR Sidelink in Release-16 and Beyond," *IEEE Commun. Stand. Mag.*, vol. 5, no. 1, pp. 12–21, Mar. 2021, doi: 10.1109/MCOMSTD.001.2000070.
9. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.
10. P. Rani *et al.*, "Federated Learning-Based Misbehaviour Detection for the 5G-Enabled Internet of Vehicles," *IEEE Trans. Consum. Electron.*, 2023.
11. Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, Mar. 2019, doi: 10.1145/3298981.
12. N. Hussain, P. Rani, N. Kumar, and M. G. Chaudhary, "A deep comprehensive research architecture, characteristics, challenges, issues, and benefits of routing protocol for vehicular ad-hoc networks," *Int. J. Distrib. Syst. Technol. IJDST*, vol. 13, no. 8, pp. 1–23, 2022.
13. T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A Federated Self-learning Anomaly Detection System for IoT," in *2019 IEEE 39th*

International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA: IEEE, Jul. 2019, pp. 756–767. doi: 10.1109/ICDCS.2019.00080.

14. Y. Liu, N. Kumar, Z. Xiong, W. Y. B. Lim, J. Kang, and D. Niyato, “Communication-Efficient Federated Learning for Anomaly Detection in Industrial Internet of Things,” in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Taipei, Taiwan: IEEE, Dec. 2020, pp. 1–6. doi: 10.1109/GLOBECOM42002.2020.9348249.
15. R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, “Fed-IIoT: A Robust Federated Malware Detection Architecture in Industrial IoT,” *IEEE Trans. Ind. Inform.*, vol. 17, no. 12, pp. 8442–8452, Dec. 2021, doi: 10.1109/TII.2020.3043458.
16. D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, “Chained Anomaly Detection Models for Federated Learning: An Intrusion Detection Case Study,” *Appl. Sci.*, vol. 8, no. 12, p. 2663, Dec. 2018, doi: 10.3390/app8122663.
17. N. Hussain and P. Rani, “Comparative studied based on attack resilient and efficient protocol with intrusion detection system based on deep neural network for vehicular system security,” in *Distributed Artificial Intelligence*, CRC Press, 2020, pp. 217–236.
18. H. Yakan, I. Fajjari, N. Aitsaadi, and C. Adjih, “A Novel AI Security Application Function of 5G Core Network for V2X C-ITS Facilities Layer,” in *ICC 2023 - IEEE International Conference on Communications*, Rome, Italy: IEEE, May 2023, pp. 6211–6217. doi: 10.1109/ICC45041.2023.10279621.
19. P. Lv, L. Xie, J. Xu, X. Wu, and T. Li, “Misbehavior Detection in Vehicular Ad Hoc Networks Based on Privacy-Preserving Federated Learning and Blockchain,” *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 4, pp. 3936–3948, Dec. 2022, doi: 10.1109/TNSM.2022.3220779.
20. J. Kamel, M. Wolf, R. W. Van Der Hei, A. Kaiser, P. Urien, and F. Kargl, “VeReMi Extension: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland: IEEE, Jun. 2020, pp. 1–6. doi: 10.1109/ICC40277.2020.9149132.
21. A. Boualouache and T. Engel, “A Survey on Machine Learning-Based Misbehavior Detection Systems for 5G and Beyond Vehicular Networks,” *IEEE Commun. Surv. Tutor.*, vol. 25, no. 2, pp. 1128–1172, 2023, doi: 10.1109/COMST.2023.3236448.
22. J. Prakash, L. Murali, N. Manikandan, N. Nagaprasad, and K. Ramaswamy, “A vehicular network based intelligent transport system for smart cities using machine learning algorithms,” *Sci. Rep.*, vol. 14, no. 1, p. 468, Jan. 2024, doi: 10.1038/s41598-023-50906-7.
23. K. Hu *et al.*, “Federated Learning: A Distributed Shared Machine Learning Method,” *Complexity*, vol. 2021, pp. 1–20, Aug. 2021, doi: 10.1155/2021/8261663.
24. L. Li, Y. Fan, M. Tse, and K.-Y. Lin, “A review of applications in federated learning,” *Comput. Ind. Eng.*, vol. 149, p. 106854, Nov. 2020, doi: 10.1016/j.cie.2020.106854.
25. V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, “Federated-Learning-Based Anomaly Detection for IoT Security Attacks,” *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2545–2554, Feb. 2022, doi: 10.1109/JIOT.2021.3077803.
26. K. Bonawitz *et al.*, “Towards federated learning at scale: System design,” *Proc. Mach. Learn. Syst.*, vol. 1, pp. 374–388, 2019.
27. T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

28. Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6G: Applications, challenges, and opportunities," *Engineering*, vol. 8, pp. 33–41, 2022.

29. T. Ryffel *et al.*, "A generic framework for privacy preserving deep learning," 2018, doi: 10.48550/ARXIV.1811.04017.

30. S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto, "Distillation-Based Semi-Supervised Federated Learning for Communication-Efficient Collaborative Training With Non-IID Private Data," *IEEE Trans. Mob. Comput.*, vol. 22, no. 1, pp. 191–205, Jan. 2023, doi: 10.1109/TMC.2021.3070013.

31. A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, May 2012, doi: 10.1016/j.cose.2011.12.012.

32. R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," *Int. J. Eng. Technol.*, vol. 7, pp. 479–482, Jan. 2018.

33. J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data," *J. Big Data*, vol. 7, no. 1, p. 104, Nov. 2020, doi: 10.1186/s40537-020-00382-x.

34. E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based Intrusion Detection System for In-Vehicle Network," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, Aug. 2018, pp. 1–6. doi: 10.1109/PST.2018.8514157.

35. S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *J. Big Data*, vol. 8, no. 1, p. 6, Dec. 2021, doi: 10.1186/s40537-020-00390-x.

36. J. Wang, M. Xu, H. Wang, and J. Zhang, "Classification of Imbalanced Data by Using the SMOTE Algorithm and Locally Linear Embedding," in *2006 8th international Conference on Signal Processing*, Guilin, China: IEEE, 2006, p. 4129201. doi: 10.1109/ICOSP.2006.345752.

37. P. Rani and R. Sharma, "Intelligent transportation system for internet of vehicles based vehicular networks for smart cities," *Comput. Electr. Eng.*, vol. 105, p. 108543, 2023.

38. Z. Khan, M. Chowdhury, M. Islam, C.-Y. Huang, and M. Rahman, "Long short-term memory neural network-based attack detection model for in-vehicle network security," *IEEE Sens. Lett.*, vol. 4, no. 6, pp. 1–4, 2020.

39. J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel Deep Learning-Enabled LSTM Autoencoder Architecture for Discovering Anomalous Events From Intelligent Transportation Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, Art. no. 7, Jul. 2021, doi: 10.1109/TITS.2020.3017882.

40. M. Abdel-Basset, N. Moustafa, H. Hawash, I. Razzak, K. M. Sallam, and O. M. Elkomy, "Federated intrusion detection in blockchain-based smart transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2523–2537, 2021.

41. M. Driss, I. Almomani, Z. e Huma, and J. Ahmad, "A federated learning framework for cyberattack detection in vehicular sensor networks," *Complex Intell. Syst.*, vol. 8, no. 5, pp. 4221–4235, 2022.