Fed-IoVIDS: Intrusion Detection based on Attack Behavior Analysis with Temporal Model on IoV Considering Privacy Protection

Rui Chen, Jing Zhao

***Abstract*—With advancements in distributed communications for the IoV, security threats expose significant challenges. While current IoV intrusion detection systems demonstrate high accuracy, they rely heavily on private or easily forged data. Moreover, the training process incurs increased network communication costs, fails to protect user privacy, and distorted data degrades detection performance. To address these limitations, we proposes a distributed federated learning-based intrusion detection model for IoV using non-private behavior features. Firstly, we design a data processing algorithm that groups and slices IoV communication messages into time series. Then, behavior vectors are extracted using an attention-based time series model designed in this work. Attacks are detected by spatially transforming the residuals with a neural network. Finally, we use a federated learning algorithm for data processing and training of the model, effectively reduce communication burden and protect privacy training data on the vehicle-side. Extensive experiments on two datasets validate the proposed model, achieving F1 scores of 91.66% and 90.25% respectively, outperforming state-of-the-art methods. We publicly release the model and algorithms to improve reproducibility and accessibility of effective IoV intrusion detection solutions.**

***Index Terms*—Distributed Communication, Attack Behavior analysis, Intrusion Detection, Privacy Protection, federated learning**

I. INTRODUCTION

In recent years, the proliferation of personal mobile devices has catalyzed advancements in the field of distributed communications, forming a major technological foundation for the Internet of Vehicles (IoV). As a promising paradigm for next-generation Vehicle-to-Everything (V2X) systems, IoV is expected to transform the underlying communication and transportation infrastructure [1]. The widespread adoption of V2X enabled by wireless technologies will likely be facilitated by IoV. In this context, a vehicle connected to a mobile network can communicate with an On-Board Unit (OBU) equipped with a Road Side Unit (RSU) using IEEE 802.11p or 5G protocols. IoV represents an integrated, open network architecture that interconnects vehicles, nearby infrastructure, and the public Internet.

However, the high level of connectivity in connected vehicles makes them vulnerable to various types of cyber attacks that could result in malicious control of the vehicle on the road, posing serious threats to human life [2]–[7]. Therefore, it is imperative to study intrusion detection models that can protect vehicle privacy data and communication units from malicious attacks. By implementing an intrusion detection model, the embedded system of the vehicle can be prompted to isolate the attacked network or enter a safe mode, reducing the security threats while the vehicle is on the road. Also, security threats to the IoV can be broadly classified into two categories: in-vehicle network (IVN) attacks and external vehicle network attacks. IVN attacks refer to attacks that occur within the vehicle network, while external vehicle network attacks are attacks that originate from outside the vehicle network. These attacks include denial of service (DoS), distributed DoS (DDoS), replay, spoofing, and brute-force attacks. DDoS attacks can be launched by flooding a node with redundant messages, causing it to overload and rendering it unable to process legitimate requests. Spoofing attacks, on the other hand, can attempt to interrupt communication between two nodes, leading to an attack similar to a DoS attack.

In prior research such as [8], autoencoder-based intrusion detection systems using Long Short-Term Memory (LSTM) networks were proposed, achieving high accuracy of 99-100% in detecting known and unknown attacks in IoV networks. While the high accuracy provides guarantees for in-vehicle network security, such performance relies on decisive weighted features in datasets, as basic neural networks can also attain high detection accuracy. Recent studies like [2], [3], [9] have applied transfer learning techniques for intrusion detection in in-vehicle networks, utilizing Controller Area Network (CAN) DATA fields from the CAN intrusion dataset [10], [11] as training features. By transforming the DATA into matrix representations and using Convolutional Neural Networks (CNNs), promising results have been achieved. However, the CAN DATA field contains extensive private vehicle information that is easily falsified and difficult to analyze, posing serious user privacy risks. Additionally, existing DATA features are generated based on attacker characteristics without considering privacy preservation or communication overheads with RSU during model training. While timestamp and ID fields can effectively simulate messages, the simulated DATA risks inauthenticity. Effective IoV intrusion detection that protects user privacy and communication efficiency remains an open challenge requiring further research.

To address the challenges prevalent in current research,

Rui Chen is with the School of Software Technology, Dalian University of Technology, DaLian, Liaoning 116024, China (e-mail: 72117004@mail.dlut.edu.cn).

Jing Zhao is with the School of Software Technology, Dalian University of Technology, DaLian, Liaoning 116024, China (e-mail: zhaoj9988@dlut.edu.cn).

we propose an intricate intrusion detection model based on attack behavior analysis. Initially, our meticulously designed data processing algorithm is capable of excluding data containing private information, such as easily forged CAN data fields. Further, in the training dataset, we selectively use non-sensitive requests like duration, bytes sent, and bytes received, thereby significantly mitigating the risk of privacy leakage induced by sensitive data. On this basis, we propose the same source property of attacks. This property groups data from the perspectives of both the attacker and the regular user, and then slices the grouped data over time to obtain a training dataset rich in behavioral information. Subsequently, we innovatively enhance the detection model by fusing the detection results of each timestep with the final result using an additive attention mechanism and employ a residual neural network for the ultimate detection. This design improves the detection outcome in both accuracy and robustness compared to existing models. Finally, we use a federated learning algorithm to train the detection model. This strategy not only protects the privacy data within the model from being leaked but also reduces the communication costs for vehicles participating in the training. In summary, our method yields significant advantages in privacy protection, communication cost reduction, and detection result enhancement.

Above all, we are the major contribution of this work is as follows:

- Our proposed detection model is the first for IoV that does not rely on private user data, easily falsified features, or features with decisive weights. By extracting temporal slices containing user behaviors from grouped data, we improve model accuracy and robustness.
- We enhance the attention-based time series model to identify more user behavior information and reduce behavior data loss. Meanwhile, federated learning is leveraged to train the model, preserving user privacy while lowering communication overhead.
- Experiments on the UNSW-NB 15 [12] and CAN-intrusion-datasets [11] are conducted, with privacy-sensitive fields discarded to avoid leaks.
- Compared to state-of-the-art models, our model achieves F1 scores of 91.66% and 90.25% respectively. Additionally, we publicly release data processing algorithms and model code to alleviate the scarcity of reproducible, effective models in this domain.

The rest of this paper is organized as follows. Section II summarizes related work. In Section III, we introduce the threat models that need to be studied in this paper. In section IV, we elaborate on the overall system design of the intrusion detection model, the newly designed datasets processing algorithm and the detection model training process. Section V and VI presents comparative experiments on the detection models. Section VII concludes the work.

II. RELATE WORK

In this section, we present the relevant work considered to identify the gaps in the proposed study. Intrusion detection in external-vehicle networks has also garnered extensive attention. Keval Doshi [13] proposed a novel anomaly-based Intrusion Detection System (IDS) capable of detecting and mitigating such emerging types of DDoS attacks in a timely manner, although it does not account for other types of attacks. Yang et al. [2] introduced a multi-layered hybrid IDS that combines signature-based and anomaly-based IDS to detect both known and unknown attacks on vehicular networks, with performance evaluation conducted on the CIC-IDS 2017 [14] dataset. Khan et al. [15] presented a multi-stage intrusion detection framework to identify intrusions from ITS and generate a lower false-positive rate. The proposed framework can automatically differentiate intrusions in real-time. It is based on a state-of-normalcy and a deep learning-centric Bi-directional Long LSTM architecture, effectively identifying intrusions from the foundational network gateways and communication networks of autonomous driving vehicles.

In the realm of intrusion detection within vehicular networks, various methods have been devised for detecting vehicle anomalies and faulty sensors. Notably, some methods proposed are of low computational expense and can detect anomalies by analyzing normal behavior patterns alone. They do not require attack data to be labeled to build a profile of normal behavior, as any deviation from normal behavioral patterns may suggest the occurrence of an anomaly. Qin et al. [10] introduced a cloud-vehicle cooperative IDS based on multi-dimensional features that addresses data heterogeneity by abstracting different vehicle data into the same feature space. Thus, datasets from different vehicles can be input into a single model for multi-classification, naturally solving the problem of model portability.

Traditional machine learning techniques have been applied for intrusion detection in vehicular networks, such as tree-structured models [16], and probabilistic data structures [4], [5], [5]. With the rise of deep learning, techniques like interpretable neural networks [17]–[19], generative adversarial networks [20], autoencoders with time series models [8], and convolutional neural networks [2], [3] have also been investigated for vehicular intrusion detection. These aim to improve detection accuracy and adaptability compared to classical methods, as well as generate realistic attack samples. However, existing work has mostly focused on CAN bus networks. With more complex automotive architectures emerging, advanced deep learning solutions tailored for heterogeneous in-vehicle environments need to be developed.

Federated learning (FL) emerged and spread to address the problem of standard deep learning solutions that are difficult to implement in privacy scenarios [21]. A central server connects multiple vehicles via 5G or IEEE 802.11p protocols to jointly train intrusion detection models [22], [23]. Unlike the centralized data collection scheme in standard deep learning, the data in federated training is widely distributed across different local devices. Also, the global server only specifies the initial training model and the associated aggregation algorithm, and does not collect any training data, which do not have direct access to the training data [24], [25]. This

approach reduces some of the costs associated with traditional centralized models. However, it also introduces issues related to bandwidth and privacy when transmitting gradient updates. To address these concerns, a combination of compression and encryption can be used to speed up model transmission while also preventing unauthorized changes. Moreover, only model-related parameters are transmitted during the communication process. The transmission of raw data and its key statistics is prohibited [26]. Currently, the standard aggregation algorithm in federation learning is FedAvg [21]. The design of this algorithm assumes that the data is uniformly distributed in each local device.

Our proposed intrusion detection system (IDS) has several advantages over existing work related to IoV intrusion detection. Firstly, our IDS is trained on non-private feature data using federated learning from publicly available datasets, which effectively avoids the leakage of user's private data. Secondly, our IDS does not rely on features with deterministic weights for feature identification, which helps to avoid model misclassification and enables more effective adaptation to real-world scenarios. Thirdly, compared to other IDSs that use machine learning and deep learning techniques, our IDS achieves better accuracy in terms of detection rate.

III. Threat Model

A. In-Vehicle Network Threat Model

The in-vehicle networks are confronted with a diverse array of security threats, encompassing a spectrum from remote attacks to local access intrusions. In the domain of remote attacks, DoS/DDoS assaults have the potential to disrupt network services, while malware and botnets could be utilized to commandeer or coordinate attacks on vehicles [6], [7]. Local attacks include physical access intrusions, such as direct assaults via a vehicle's OBD-II port, and sensor spoofing attacks that manipulate vehicle behavior through deceptive signals like falsified GPS data. In the realm of wireless attacks, man-in-the-middle attacks have the capability to intercept and alter communication data, whereas electronic jamming and signal blocking can undermine the wireless communications of a vehicle. These security threats not only jeopardize the safety of vehicles and their passengers but also pose a risk to driver privacy and can negatively impact the brand reputation of vehicle manufacturers. Consequently, as vehicles increasingly rely on interconnected electronic systems, the implementation of multi-layered, comprehensive intrusion detection measures becomes particularly critical.

B. External-Vehicle Network Threat Model

In similar contexts, external-vehicle networking technologies can facilitate the interaction and communication between vehicles and other intelligent transportation system entities, including pedestrians, infrastructure, smart terminals, and network systems [2]–[5]. With the advancement of modern vehicular networking technologies, the connectivity between vehicles is increasingly enhanced, and the external vehicular networks are gradually evolving into a comprehensive large-scale system encompassing many networks and devices. Therefore, such peripheral vehicular networks are extremely vulnerable to common network threats, where each vehicle and device can be a potential attack entry point. Typical network attack means in vehicular network environments include Denial of Service (DoS), Global Positioning System (GPS) spoofing, signal jamming, data sniffing, brute-force cracking, zombie network control, system penetration, and cyber attacks. Data privacy attacks within external-vehicle network involve unauthorized data access and dissemination, as well as identity theft, which could lead to the exposure of personal user information or impersonation of user identities.

To enhance the intrusion detection capabilities of intelligent vehicular network systems and prevent severe damages that could be caused by hacker attacks, we propose an intrusion detection model for vehicular networks based on distributed federated learning using non-proprietary behavioral features.

IV. Proposed Method

This section analyzes the design rationale and mathematical derivations of the data processing algorithm from the perspective of attack behaviors, and introduces the application scenarios of the algorithm. From Fig. 1, to prevent leakage of private user data, we avoid using features containing privacy information. Instead, we adopt behavior analysis techniques to enhance detection capabilities. Additionally, we elaborate on the distributed training solution to address the issues of communication overhead on in-vehicle devices and potential data leakage risks with centralized training.

A. Data Preprocessing

1) ***The Same Source Property****:* Fig. 2 illustrates mining potential attack behaviors using the same source property. Inspired by cache algorithms where users often reaccess recently visited areas, we found through analysis that users tend to revisit addresses within a short timeframe, termed the same source property.

In external/in-vehicle network attacks (Fig. 2A), attackers repeatedly target addresses to achieve goals, often disorderly. Even deliberately disguised, other features may expose them. In contrast, normal users show contextual relationships and do not redundantly message vehicles. The same source property also manifests in in-vehicle networks (Fig. 2B). For instance, brake/acceleration ECUs are invoked more frequently than sunroof/windows during driving.

From Algorithm 1, we grouped datasets by the same source property, using IP source address for vehicle extranets and CAN ID for intranets. We stored the grouped data in files by line with the same grouping flag to avoid excessive memory usage during calculation and enable easy viewing of grouping results and error checking.

2) ***Extraction Behavior Vectors****:* The grouped data cannot be directly used for model training, we need further processing. In this article, we used the UNSW-NB 15 [12] and CAN-intrusion-datasets [11]. In the UNSW-NB 15 datasets, only *dur*, *sbytes*, and *dbytes* features are retained , in the

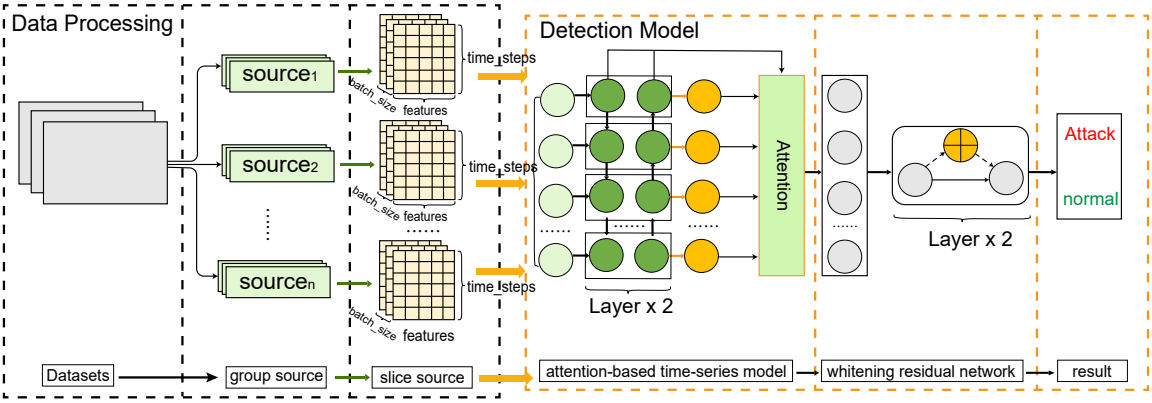

Fig. 1. Intrusion detection model system design diagram, including the process of data processing, model design and vector transformation. Data Processing: Schematic diagram of the datasets processing flow of the detection model, Detection Model: Identify the user's behavior from the sliced data to derive the decision result.

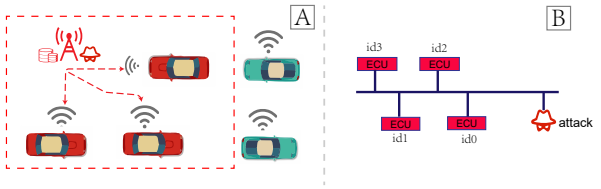

Fig. 2. A: The *same source property* of the attacker on the External-Vehicle Network. B: The *same source property* of the attacker on the In-Vehicle Network.

Algorithm 1 Data Processing Algorithms Based on The Same Source Property

Input: D: External/In-Vehicle network communication datasets

Output: x_{norm}: A datasets in vector form containing behavioral information available for training

1: **for** D_i in D **do**
2: Format each D_i
3: Remove fields containing private information to obtain **Table I**
4: Fill in zeros for meaningless values in D_i
5: **end for**
6: $D_g \leftarrow$ Group D by unique identifiers.
7: $V \leftarrow$ Sort by the same source property and slice D_g based on the time of occurrence.
8: $x_{norm} \leftarrow \frac{V_x - V_{min}}{V_{max} - V_{min}}$. Normalize V using V_{min} and V_{max}.
9: **Return** x_{norm}

CAN-intrusion-datasets datasets, the features of *CAN ID* are dropped. Through this process, we delete the possibility of private data to avoid leakage of user private data, and these data are relatively easy to obtain. After the private data is removed, in order to facilitate better training of the time model, and then segment it, we choose 64 time steps as a segment unit, and the segmented data is used as the basic unit for training corpus.

The grouped data needs to be arranged in the order of access time, and the time step is used as the slice length to divide the grouped data. This operation retains the chronological nature of the data and is also conducive to the training of the time model. After visiting for a period of time, the user will stay at different addresses, and these stay times include the user's behavior.

To meet the requirements of the input format in the neural network and to avoid differences in the value of data, it is necessary to compress the size ratio of the value within a range, and then use the Min-Max technique to normalize the data $x_{norm} = (V_x - V_{min})/(V_{max} - V_{min})$. Where V_{min} is the smallest data in a slice, V_{max} is the largest data in a slice, V_x in a slice needs to be normalized data, x_{norm} is the normalized data. If the maximum value and the minimum value are equal, in order to avoid the denominator being 0, x_{norm} is directly set to 0.

B. Detection Model Design

Through the processing of Section IV-A, we obtained slice data, which contains user behavior information, but does not contain user privacy data. Since the user's behavior has context, we use a bidirectional recurrent neural network (Bi-LSTM) as the time series model. The output of the hidden layer of the time series model also contains a large amount of behavioral information. If it is discarded, the behavioral information will be lost, as shown in Fig. 1 Data Processing.

To avoid this problem, we improve additive attention by fusing the hidden and output layers to obtain detection vectors. Through the residual neural network, we obtained the detection results. Bi-LSTM is the abbreviation of Bi-directional Long Short-Term Memory, which is a bidirectional recurrent neural network composed of forward LSTM and backward LSTM. Compared with the unidirectional LSTM, Bi-LSTM can recognize the bidirectional information of the sequence, and can more effectively recognize the context of the sequence.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\hat{C}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \tag{4}$$

In our work, the slice order x of the time series is introduced as input to the Bi-LSTM model, and the length of the slice corresponds to the time step t in the time series model. During the initiation of the model computation, the hidden layer lacks data, necessitating the initialization of the hidden layer data to 0. Since the model in this study is tailored for different datasets, each with a distinct number of features, we must convert these different features into the same dimensionality via a vector using $ReLu(XW + b)$. Here, $W \in R^{features\_size \times hidden\_size}$, x is the data of the dataset comprising varying numbers of features as input, and $features\_size$ is the size of the feature number for different datasets.

As shown in Equation (1), the slice vectors x encapsulating behavioral data are transformed into the hidden dimensionality. Within the time series model, the previous cell state and memory gate values are computed. For example, the forgetting gate $f_t \in R^{batch\_size \times hidden\_size}$ has an output dimension set by the hyperparameters $batch\_size$ and $hidden\_size$, with a sigmoid activation. i_t, $\hat{C}_t$, and $\hat{C}_t$ follow similar formulations.

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = O_t * tanh(C_t) \tag{6}$$

After completing the calculation of the internal parameters in the time series model and unifying the dimensions, the internal results are output, such as the equation (4) outputs the cell state as the input of the cell state of the next time step, and after the final output of the time series model, we discard it not into the calculation of the attention model, the output results of the input layer, as equation (5), and the hidden layer, as equation (6), as the attention query, and the key and value.

$$scores = Softmax[tanh(q + k)W + b] \tag{7}$$

$$O_b = scores \cdot v \tag{8}$$

We hypothesize that the hidden layer outputs of the time series model may also encapsulate behavioral data patterns embedded in the time series input. To fully utilize these potential latent behavioral representations, as formalized in Equation (7), we first perform an additive operation on the query and key vectors, followed by a linear transformation and softmax normalization to obtain attention scores. In Equation (7), q corresponds to the output O_t, and k, v correspond to the hidden states h_t. The dot product of v and the $scores$ in Equation (7) fuses the information from the hidden and output layers, capturing latent user behaviors. This fused representation then passes through the residual neural network blocks in Equations $v_o = ReLu(O_b W + b_{ob})$ and $result = dropout(v_o) + v_o$ to obtain the final detection outcome.

Algorithm 2 Model Training Algorithm Based on Federated Learning

Input: N: The set of clients; T: The total number of global iterations; s: the maximum number of selected clients in each iteration

Output: w_{t+1}: Aggregated model global parameters

1: Randomly initialize global parameters w
2: **for** t ← 0 to T-1 **do**
3: Randomly choose at most s clients
4: **for** i ← 0 to s **do**
5: $g_{i,t} \leftarrow argmin_w L(D_i; w_t)$ \\ D_i represents the data of the i-th client
6: $g_t^s \leftarrow g_t^s + g_{i,t}$
7: **end for**
8: $w_{t+1} \leftarrow w_t - \eta_t \frac{|D_i|}{|D|} g_t^s$ \\The server aggregates the returned parameters
9: **end for**

C. Federated Learning For Model Training

Upon finalizing the architecture of our detection model, we embarked on the training phase using federated learning, the details of which are outlined in Algorithm 2. The procedure unfolds as follows:

(1) A central server commences by initializing a blank slate for the global model.

(2) For each round of federated learning, the central server dispatches the current global model parameters, denoted by w, to the RSU over secure dedicated lines. Subsequently, the RSU disseminates the model parameters to each vehicle client within its local network.

(3) Vehicle clients, each possessing their local dataset D_i, proceed to train the received model. They calculate updated model gradients $g_{i,t}$ and transmit these updates back to the RSU.

(4) Upon receipt, the RSU compiles the updated models from its network of vehicle clients and conveys this aggregated information, g_t^s, to the central server. The central server then applies a global aggregation algorithm to merge the updates from all RSUs, resulting in an updated global model, represented by w_{t+1}. Finally, the central server refreshes its global model with w_{t+1}, completing the current round of aggregation.

This iterative process is repeated across multiple rounds to progressively refine the model, enhancing its sophistication and detection capabilities. Each round propels the global model closer to optimal performance, leveraging the distributed learning paradigm inherent in federated learning.

TABLE I
EXPLANATION OF THE FEATURES USED IN THE PAPER

data-set	features	explanation
UNSW-NB 15 [12]	srcip	Source IP address, as a condition for data grouping
	sbytes	Source to destination transaction bytes
	dbytes	Destination to source transaction bytes
	dur	The time spent visiting a destination address.
CAN-intrusion-datasets [11]	Timestamp	recorded time, calculating *dur* features by it
	dur	The time spent visiting a destination address.
	CAN ID	identifier of CAN message in HEX (ex. 043f) , as a condition for data grouping
	DLC	number of data bytes, from 0 to 8

TABLE II
FEDERATED LEARNING FOR TRAINING MODEL PARAMETERS AND HYPERPARAMETERS

Round number	150
Client number	100
Number of clients selected for a round	10
Local clients batch size	10
Local clients epoch	5
Learning rate	0.015
Learning rate scheduler	0.95
optimization function	Adam
loss function	Cross Entropy Loss

V. EXPERIMENTATION

A. Datasets

To rigorously assess the detection capabilities of our model under a variety of attack scenarios, we incorporated two extensively recognized datasets, namely UNSW-NB15 [12] and CAN-Intrusion-Dataset [11], originating from different settings to provide a comprehensive performance evaluation.

The UNSW-NB15 dataset comprises a substantial compilation of 2,540,047 events, out of which 321,283 are identified as attack instances. Additionally, the CAN-Intrusion-Dataset serves as a critical benchmark for gauging the efficacy of in-vehicle network intrusion detection systems. This dataset encompasses a total of 4,613,909 records, with attacks constituting 2,244,041 of these entries.

As indicated in Table I, the selection of features for model training was meticulously derived using a specialized data processing algorithm, ensuring that the most relevant and impactful attributes were included in the analysis.

B. Experimental Setup

This study's algorithm was implemented in a Ubuntu/Linux 18.04 OS environment, with Python 3.10 as the programming language of choice. The development of the deep neural network models leveraged the application programming interfaces (APIs) of the PyTorch 1.13 deep learning framework. Model development and initial experimentation were carried out on a robust computing platform, which boasted an Nvidia RTX 3060Ti GPU, an 8-core Intel Core i7 10700F CPU running at 2.9GHz, and 32GB RAM.

Model training was GPU-accelerated, utilizing the Adam optimization algorithm across a total of 150 epochs within a federated learning context. In this setup, each federated learning participant completed 10 epochs of local training. The model's initial learning rate was set at 0.015 and underwent an exponential decay after each epoch, decreasing by a factor of 0.95, to fine-tune the learning process. Detailed parameters underpinning the federated learning approach are delineated in Table II.

C. Evaluation Metrics

This paper selects the F1 Score as the evaluation metric. The F1 Score is a highly significant performance measure because it does not exhibit bias towards any particular class, unlike precision or recall, but rather provides a comprehensive performance index. The F1 Score is a commonly used metric in statistics and machine learning to assess the accuracy of classification models, particularly in scenarios where there is an imbalance in the dataset. It represents the harmonic mean of precision and recall.

VI. EVALUATION AND DISCUSSION

A. Communications Cost and The Same Source Property Validation Experiments

In the context of external and in-vehicle networks, Road-Side Units (RSUs) and vehicles wirelessly engage in communication via 5G or 802.11p standards to facilitate federated learning. Centralized training incurs significant overhead due to the necessity of uploading local datasets to central repositories during vehicle operations. We evaluate these costs by examining the data transfer between local vehicles and the RSU.

Fig. 3B illustrates that the costs associated with centralized training are proportional to the size of the dataset, leading to substantial overhead for larger datasets. In contrast, federated learning is more efficient, as it necessitates only intermittent uploads of model weights. In our experimental setup, the cost of a single round of federated uploads is determined by the cumulative size of the local models and the dataset attributes. Conversely, the download costs are equivalent to the size of the global model disseminated to all participating vehicles. We quantified the aggregate data exchanged over 450 communication rounds between the vehicles and the RSU. For federated learning, the communication cost is computed as $cost = \sum_{i}^{round} W_s(i)$, where $round$ signifies the number of communication iterations and W_s represents the size of the model weights in bits. Centralized training costs, on the other hand, equate to the total size of the dataset since it requires a central aggregation of data from all vehicles. The

data indicates that federated learning generally results in lower communication overhead.

Our research also encompassed four experimental validations of the effectiveness of the proposed "user behavior same source property." When the model was trained without grouping by this property, we observed considerable variations and instability over fixed 64-record intervals. Control experiments, which did not utilize attention mechanisms, confirmed that the observed jitter was associated with the time series data and the attention mechanisms employed. These findings support the conclusion that incorporating the "same source property" improves the stability and convergence of the training process.

B. Ablation Study of Modules

To ascertain the necessity and efficiency of the modular components within the model introduced in this study, we undertook an exhaustive ablation study. This involved systematically dismantling individual modules and conducting quintuple sets of control experiments per module, adopting the median outcome as a stabilizing factor against variabilities in the experimental conditions.

Depicted in Fig. 3A, our objective was to discern the contributory significance of each module by either substituting or excluding them and observing the resultant impact on the model's performance in detection tasks. Our investigative efforts extended to the analysis of time series data using LSTM and attention-augmented Bi-LSTM structures, wherein the Bi-LSTM's efficacy markedly exceeded that of the standard unidirectional LSTM model. In scenarios involving unidirectional time series, control experiments were implemented both with and without the integration of attention mechanisms. These trials demonstrated a clear superiority of the attention-enhanced unidirectional time series over its counterpart lacking such mechanisms.

Furthermore, the training process exhibited significant volatility in the absence of the "same source property" within the data processing algorithm. This observation led us to conclude that the proposed algorithm for data aggregation and the model's modules are essential, effectively mitigating the risk associated with modular redundancy. Given the array of evaluation metrics, the Bi-LSTM model with time series analysis emerged as the superior choice, outclassing its contemporaries in the realm of detection capabilities.

C. Comparisons With The State of The Art

We have validated the data processing algorithm and detection model proposed in this paper on two datasets. From Table III, during the process of handling the datasets, we observed that it contained a considerable number of deterministic weighted features in Table I. Our findings indicate that, even when utilizing more streamlined models and data processing techniques, this study attains high levels of accuracy and robust detection performance, offering a favorable comparison to the results [17], [28], [29]. However, the datasets also contained elements of personal data. To address privacy concerns and adhere to data protection standards, we deliberately excluded any features that could potentially contain sensitive personal information or exhibit deterministic weighting.

TABLE III
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART MODELS

Model	F1	privacy-sensitive fields	Precision	Privacy Protection
Proposed Model	**91.66%, 90.25%**	**w/o, w/o**	**90.16%, 94.71%**	**YES**
CNN-based IDS [27]	98.83%	w/	99.10%	NO
Homogenous DeepRed [17]	92.2%-96.32%	w/	89.42% - 98.32%	NO
Enhance-MS -IDS [15]	-	w/	99.13%	NO
WCGAN-IDS [28]	-	w/	86.3%	NO
MTH-IDS [2]	99.895%	w/	-	NO
FedMix [29]	91.3%	w/	98.5%	NO

In previous studies [17] [30], explainable deep learning technologies were used for model detection, achieving an F1 score of 0.90-0.97. In our study, we used the same datasets and achieved an F1 score of 91.66%. However, their work overlooked privacy protection and the risk of data leakage through model training. Other research [2], [27] used deep learning for intrusion detection but did not consider the communication pressure and privacy issues in edge training [29]. In contrast, our approach integrates federated deep learning with their training methods, significantly reducing communication costs compared to centralized training on a central server, and is not limited by the size of the data set.

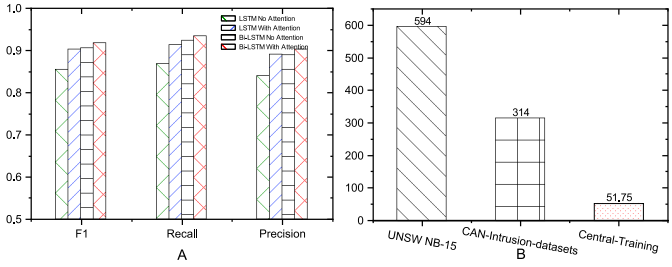

Fig. 3. A: Ablation Study of Modules on Different Datasets. B: Communication Costs on Different Datasets

In the CAN-intrusion-datasets, the DATA field is where user payloads are stored in CAN communication data, which contains users' command operations and other private data. In real scenarios, these private data are difficult for security personnel to analyze, so this feature was not considered in our model. We performed experimental training with three fields in the CAN-intrusion-datasets datasets. Compared to other studies [2], we found that if this field is removed, other studies would not be able to proceed, but our model can achieve an F1 value of 91.66%.

VII. CONCLUSION

We proposed a federated learning-based vehicular network intrusion detection model using non-private behavioral features, outperforming current methods. Our approach includes a data processing algorithm that converted vehicular network messages into time series, and an attention-based model for behavioral vector extraction. Neural networks were used for detecting intrusions through spatial transformations of residuals. This method reduced communication load and ensures data privacy. Validation on two datasets yielded F1 scores of 91.66% and 90.25%, surpassing other methods. We are open-sourcing our model for better accessibility and repeatability.

Given the scarcity of labeled datasets in this domain, we intend to extend this work by reducing model dependence on limited labeled data. With the assistance of generative adversarial algorithms and game theory, we can potentially learn the underlying distributions of positive and negative data from the available datasets. This would mitigate the issue of missing data and inadequate labeling during model training and detection.

REFERENCES

[1] E. Moradi-Pari, D. Tian, M. Bahramgiri, S. Rajab, and S. Bai, "Dsrc versus lte-v2x: Empirical performance analysis of direct vehicular communication technologies," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[2] L. Yang, A. Moubayed, and A. Shami, "Mth-ids: A multitiered hybrid intrusion detection system for internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 616–632, 2021.

[3] L. Yang and A. Shami, "A transfer learning and optimized cnn based intrusion detection system for internet of vehicles," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 2774–2779.

[4] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, p. 100198, 2020.

[5] S. Garg, A. Singh, G. S. Aujla, S. Kaur, S. Batra, and N. Kumar, "A probabilistic data structures-based anomaly detection scheme for software-defined internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3557–3566, 2020.

[6] G. X. W. Wu, R. Li, "A survey of intrusion detection for in-vehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 919 – 933, 2019.

[7] S. Rajapaksha, H. Kalutarage, M. O. Al-Kadri, A. Petrovski, G. Madzudzo, and M. Cheah, "Ai-based intrusion detection systems for in-vehicle networks: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–40, 2023.

[8] 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 Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4507–4518, 2020.

[9] I. Ahmed, G. Jeon, and A. Ahmad, "Deep learning-based intrusion detection system for internet of vehicles," *IEEE Consumer Electronics Magazine*, vol. 12, no. 1, pp. 117–123, 2021.

[10] J. Qin, Y. Xun, and J. Liu, "Cvmids: Cloud-vehicle collaborative intrusion detection system for internet-of-vehicles," *IEEE Internet of Things Journal*, 2023.

[11] H. Lee, S. H. Jeong, and H. K. Kim, "Otids: A novel intrusion detection system for in-vehicle network by using remote frame," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2017, pp. 57–5709.

[12] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[13] K. Doshi, Y. Yilmaz, and S. Uludag, "Timely detection and mitigation of stealthy ddos attacks via iot networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2164–2176, 2021.

[14] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, pp. 108–116, 2018.

[15] I. A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, and A. Jolfaei, "An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 469–25 478, 2021.

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

[17] S. Almutlaq, A. Derhab, M. M. Hassan, and K. Kaur, "Two-stage intrusion detection system in intelligent transportation systems using rule extraction methods from deep neural networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[18] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PloS one*, vol. 11, no. 6, p. e0155781, 2016.

[19] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2016, pp. 130–139.

[20] G. Xie, L. T. Yang, Y. Yang, H. Luo, R. Li, and M. Alazab, "Threat analysis for automotive can networks: A gan model-based intrusion detection technique," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4467–4477, 2021.

[21] 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.

[22] Z. Qu, Y. Tang, G. Muhammad, and P. Tiwari, "Privacy protection in intelligent vehicle networking: A novel federated learning algorithm based on information fusion," *Information Fusion*, vol. 98, p. 101824, 2023.

[23] F. Liang, Q. Yang, R. Liu, J. Wang, K. Sato, and J. Guo, "Semi-synchronous federated learning protocol with dynamic aggregation in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4677–4691, 2022.

[24] L. Xing, P. Zhao, J. Gao, H. Wu, and H. Ma, "A survey of the social internet of vehicles: Secure data issues, solutions, and federated learning," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 2, pp. 70–84, 2022.

[25] H. Zhou, Y. Zheng, H. Huang, J. Shu, and X. Jia, "Toward robust hierarchical federated learning in internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[26] X. Cao, T. Başar, S. Diggavi, Y. C. Eldar, K. B. Letaief, H. V. Poor, and J. Zhang, "Communication-efficient distributed learning: An overview," *IEEE Journal on Selected Areas in Communications*, 2023.

[27] A. Oseni, N. Moustafa, G. Creech, N. Sohrabi, A. Strelzoff, Z. Tari, and I. Linkov, "An explainable deep learning framework for resilient intrusion detection in iot-enabled transportation networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[28] Y. He, M. Kong, C. Du, D. Yao, and M. Yu, "Communication security analysis of intelligent transportation system using 5g internet of things from the perspective of big data," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[29] J. Zhao and R. Wang, "Fedmix: A sybil attack detection system considering cross-layer information fusion and privacy protection," in *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2022, pp. 199–207.

[30] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, and N. Kumar, "P2sf-iov: A privacy-preservation-based secured framework for internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 22 571–22 582, 2021.