IoV-BERT-IDS: Hybrid Network Intrusion Detection System in IoV Using Large Language Models

Mengyi Fu, *Graduate Student Member, IEEE*, Pan Wang, *Member, IEEE*, Minyao Liu, Ze Zhang, and Xiaokang Zhou, *Member, IEEE*

***Abstract*—The traditional vehicular ad hoc network (VANET) gradually evolved into the Internet of Vehicles (IoV), which has also become a potential target for attacks and faces security challenges in an open network environment. Intrusion detection systems (IDS) based on machine learning (ML) and deep learning (DL) are introduced to mitigate security threats. However, existing ML/DL-based IDS suffer from challenges in IoV environments. First, due to the limitations of ML/DL-based methods, classification performance is unsatisfactory when they extract only unidirectional contextual features or spatial characteristics. Second, existing research on in-vehicle network IDS often limits validation and testing to a static dataset of a single vehicle model. This approach may not adequately address diverse potential attacks in a dynamic environment. Third, few studies of hybrid IDS can simultaneously implement in-vehicle and extra-vehicle network intrusion detection. Large language models (LLM) have shown outstanding applications in fields such as natural language processing (NLP) and computer vision (CV). In particular, bidirectional encoder representations from transformers (BERT) obtain new state-of-the-art results on eleven famous NLP tasks. Consequently, this paper introduces a hybrid network IDS in IoV utilising LLM, denoted as IoV-BERT-IDS. This framework encompasses four modules: semantic extractor (SE), input embedding, IoV-BERT-IDS pre-training, and IoV-BERT-IDS fine-tuning. To conform to the BERT model, the semantic extractor is introduced to transform traffic data devoid of apparent semantics into contextual semantics, comprising bidirectional and unidirectional SE. Through SE, controller area network (CAN) data is transformed into a CAN byte sentence (CBS), while extra-vehicle network traffic data is transformed into a traffic byte sentence (TBS). Additionally, two pre-training tasks, the masked byte word model (MBWM) and next byte sentence prediction (NBSP) are proposed to acquire bidirectional contextual features from contextual semantics. These features can be adapted to downstream tasks in both in-vehicle and extra-vehicle networks through fine-tuning. Experiments demonstrate that IoV-BERT-IDS outperforms in CICIDS, BoT-IoT, Car-Hacking, and In-vehicle network intrusion detection challenge (IVN-IDS) datasets and shows good generalisation capabilities to in-vehicle networks of different vehicles.**

***Index Terms*—BERT, intrusion detection system, Internet of Vehicle, large language model, pre-training model.**

I. Introduction

With the increasing demand for road and in-vehicle status monitoring, smart connected cars are equipped with intelligent systems, remote communication applications, and sensors. By combining smart connected vehicles [1] with the new generation of information and communication technologies, the traditional Vehicular Ad-hoc Network (VANET) has gradually evolved into the Internet of Vehicles (IoV) [2]. As shown in Fig. 1, the IoV can be divided into two main components: the extra-vehicle and in-vehicle networks. Extra-vehicle network refers to Vehicle-to-Everything (V2X), covering Vehicle-to-Vehicle (V2V) [3], Vehicle-to-Infrastructure (V2I) [4], Vehicle-to-Network (V2N), and Vehicle-to-Pedestrian (V2P). In contrast, the in-vehicle network [5] refers to the network structure inside the vehicle, which contains various hardware, subsystems, and their communications inside the vehicle. In a narrow sense, the extra-vehicle network mentioned in this paper focuses on the traditional internet network, and the in-vehicle network refers to the in-vehicular CAN bus network. Multiple network connections are created between different types of terminals, and multiple transmission protocols exist. Under terminal quantisation, network heterogeneity, and connection diversification in IoV, vehicle-road-cloud messaging may be eavesdropped, tampered with, and blocked, resulting in privacy leakage of multiple parties and even causing serious traffic accidents [6]. Intrusion Detection Systems (IDS) can actively monitor network traffic without affecting network performance, detect real-time attacks, and respond to changing networks.

IDSs based on Machine Learning (ML) and Deep Learning (DL) have powerful mass data processing capability and unknown attack detection capability [7], which makes them the mainstreams to solve the security challenges of IoV. However, there are significant differences in the encoding rules for the data and arbitration segments used by the Controller Area Network (CAN) bus [8] within the in-vehicle network. Additionally, as network services evolve, the network protocols used in the extra-vehicle network constantly expand. The syntax structures of headers and payload sections in different network traffic also

Manuscript received 11 January 2024; revised 24 April 2024; accepted 27 April 2024. Date of publication 17 May 2024; date of current version 14 February 2025. This work was supported in part by Future Network Innovation Research and Application Projects under Grant 2021FNA02006, in part by the Development of an Ultra-large-scale Ubiquitous Network Quality Monitoring System Based on Trusted Edge Intelligence under Grant SYG202311, and in part by the Foundation of State Key Laboratory of Public Big Data under Grant PBD2022-10. The review of this article was coordinated by the Guest Editors of the Special Section on Large Models for Future Vehicles and Transportation. *(Corresponding author: Pan Wang.)*

Mengyi Fu, Pan Wang, Minyao Liu, and Ze Zhang are with the School of Modern Posts, Nanjing University of Post and Telecommunications, Nanjing 210003, China (e-mail: 2023070802@njupt.edu.cn; wangpan@njupt.edu.cn; 1222097606@njupt.edu.cn; 1022072003@njupt.edu.cn).

Xiaokang Zhou was with the Faculty of Data Science, Shiga University, Hikone 522-8522, Japan, and also with the RIKEN Center for Advanced Intelligence Project, RIKEN, Tokyo 103-0027, Japan. He is now with the Faculty of Business Data Science, Kansai University, Osaka 565-0823, Japan (e-mail: zhou@kansai-u.ac.jp).

Digital Object Identifier 10.1109/TVT.2024.3402366

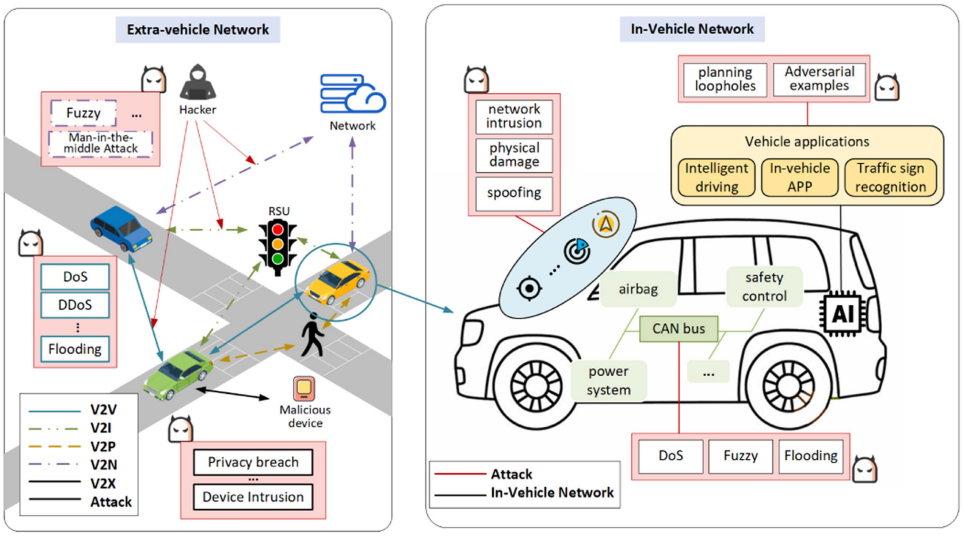


Fig. 1. Extra-vehicle network vs. in-vehicle network.

exhibit notable variations. Consequently, these models demonstrate insufficient generalisation abilities in intrusion detection within the IoV. Researchers such as Pan Wang have proposed the method ByteSGAN [9], which utilises Generative Adversarial Networks (GAN) to resample and generate minority-class samples for data augmentation. Although these techniques enhance data diversity through resampling, the resulting augmented samples may not accurately represent the true traffic distribution, failing to address the model's generalisation issue fundamentally. Huoh et al. [10] introduced an approach combining the original byte-level features with meta-features related to traffic to leverage the relationships between packets fully. These features are inputs to a Graph Neural Network (GNN) to enhance the model's generalisation performance. While this method improves feature representation and extraction from a structural perspective, it still relies on a substantial number of labelled samples for fine-tuning. In scenarios where labelled samples are scarce, achieving effective model generalisation remains challenging.

Large-scale language models have recently entered an era of rapid development in natural language processing, computer vision, and other fields. Google's BERT [11] first proved the strong potential of pre-trained models; OpenAI's GPT series, Anthropic's Claude and so on, continue to explore the boundaries of language models. Motivated by ET-BERT [12], which has performed well on multiple benchmarks in natural language processing, BERT remains a suitable solution. Therefore, this paper attempts to apply BERT to the field of IoV for traffic intrusion detection. The goal is to enhance model generalisation under the constraint of limited labelled samples by leveraging massive unlabeled pre-training and fine-tuning with a small amount of labelled data.

However, considering the unique characteristics of raw traffic input and network task output, existing NLP pre-training models cannot be directly applied to the domain of network traffic. Constructing a BERT pre-training model for network traffic faces several challenges:

1) During the pre-training process, integrating semantic information effectively is challenging due to the heterogeneous headers and payloads in traffic data.
2) The lack of uniformity in protocols between in-vehicle networks and extra-vehicle networks makes it difficult to align features from both types of traffic. Encoding multi-modal IoV traffic into a common semantic space becomes a challenge.
3) Designing effective pre-training tasks to achieve contextual understanding and capture bidirectional context features in traffic data is essential for addressing the context dependency in traffic classification tasks.

This paper contributes in the following aspects:

1) This paper introduces IoV-BERT-IDS, a hybrid network traffic IDS designed for both in-vehicle and extra-vehicle networks, leveraging the BERT model. By representing raw traffic data from both networks as hexadecimal strings called Byte Sentences (BS, introduced in Section III), IoV-BERT-IDS can learn a generic representation of traffic and be fine-tuned to address intrusion detection tasks for both networks individually.
2) IoV-BERT-IDS proposes a novel data preprocessing method called Semantic Extractor. This method addresses the challenge of processing traffic data with ambiguous semantics, transforming it into contextual semantic traffic pairs. It comprises two components as in Fig. 2: unidirectional SE processes unlabeled packet data as USL for fine-tuning, while bidirectional SE extracts labelled packet pairs as BSL for pre-training.
3) The paper introduces two pre-training tasks: the Masked Byte Word Model (MBWM) and the Next Byte Sentence Prediction (NBSP). These tasks aim to learn bidirectional contextual features from contextual semantics.
4) The model is validated using diverse datasets, including CICIDS [13], BoT-IoT [14], Car-Hacking [15], and In-vehicle network intrusion detection challenge (IVN-IDS) datasets. Experimental results showcase the model's

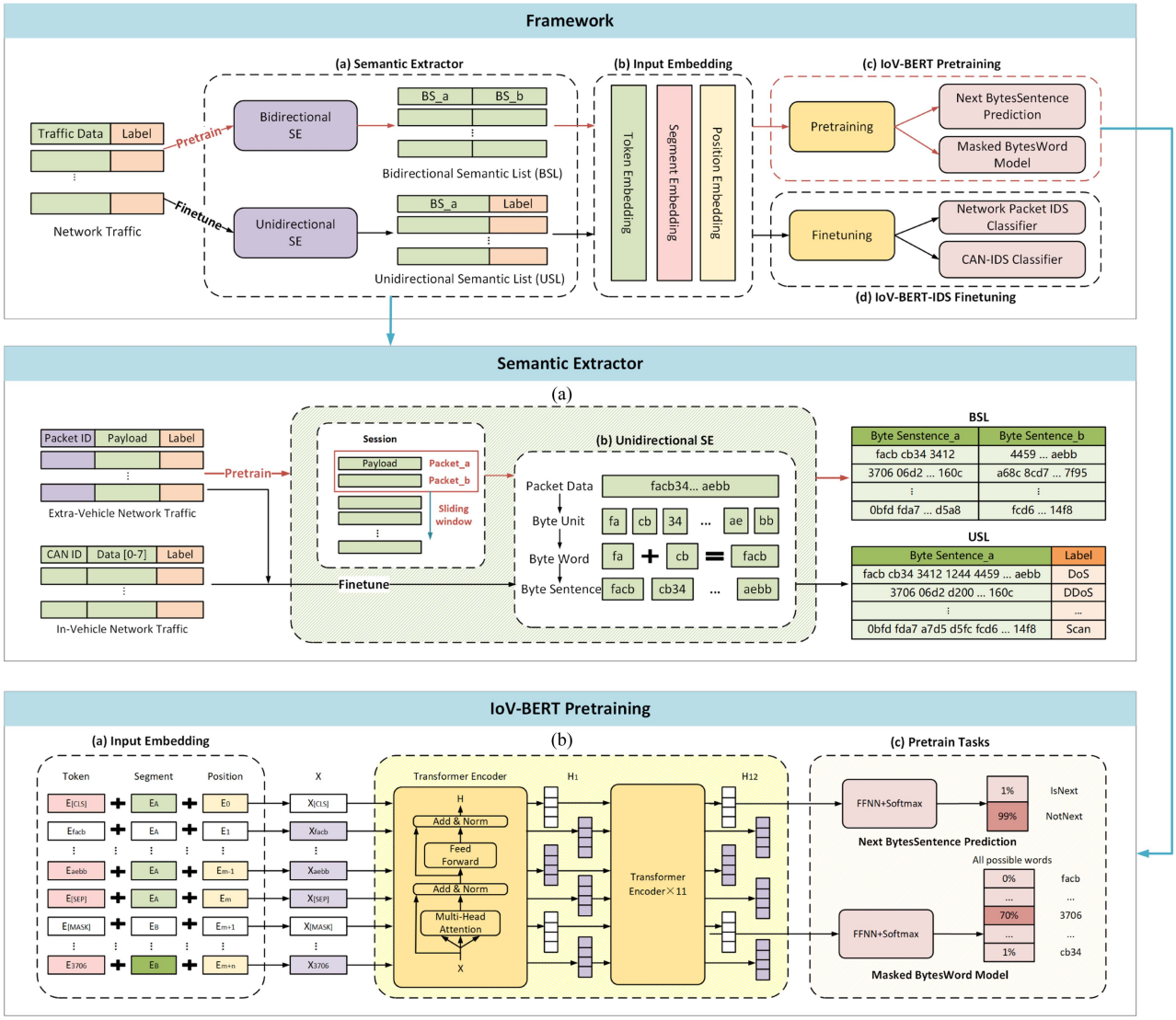


Fig. 2. The overall framework of IoV-BERT-IDS. Semantic Extractor: (a) Bidirectional SE. IoV_BERT pretraining: (a) Input embedding. (b) IoV-BERT. (c) Pretrain tasks.

high intrusion detection accuracy for in-vehicle and extra-vehicle networks. Notably, the model demonstrates well generalisation across various in-vehicle networks of different vehicles.

The chapter organisation of this paper is as follows: Section I is an overall introduction; Section II is related research works; Section III is the preliminary for the subsequent sections; Section IV presents the framework and description for IoV-BERT-IDS; Section V illustrates the experimental setup including the dataset, experimental design, and evaluation metrics.; Section VI is the evaluation of IoV-BERT-IDS through experiments; Section VII is the conclusion.

II. Related Works

A. Intrusion Detection in Internet of Vehicles

The intrusion detection system in IoV monitors vehicle networks' real-time status. Its capabilities include identifying potential security threats like hacker attacks and virus infections and implementing necessary defensive measures to uphold vehicle safety [16]. Li et al. [17] developed an IDS with a Convolutional Neural Network (CNN), leveraging transfer and ensemble learning, achieving over 99.25% detection rates but struggled with temporal relationships and feature representation. Alladi et al. [18] presented an AI-based intrusion detection architecture utilizing Deep Learning Engines (DLE) on Multi-access Edge Computing (MEC) servers for cybersecurity in the IoV network. Nie et al. [19] employed CNN to analyze RSU link load behaviors against various attacks. Yu at al. [20] constructed a Federated Long Short-term Memory (Federated-LSTM) based IDS, based on the periodicity of in-vehicle network message ID sequence. Li et al. [21] proposed a Multitiered Hybrid IDS (MTH-IDS) that incorporates signature-based and anomaly-based IDS, achieving high F1-scores for the zero-day attack detection on vehicular networks. With the increasing integration of Electronic Control Units (ECUs) and the lack of encryption and authentication in CAN, Alkhatib et al. [22] introduced CAN-BERT, a

DL-based IDS that utilises BERT to learn CAN bus arbitration identifier sequences for detecting cyber attack, addressing the vulnerability of ECUs and unencrypted CAN communication.

B. Pre-Training Model

In natural language processing, the deep bidirectional pre-training model based on Transformers achieves the best results for multiple tasks. With this representation type and structure, Li et al. [23] presented a Unified Pre-trained Language Model (UniLM) that was achieved by employing a shared transformer network and utilising specific self-attention masks to control what context the prediction conditions on. In sequence-to-sequence tasks, UniLM manages context exposure differently for its encoder and decoder, thus effectively controlling context under diverse prediction conditions, enhancing the model's versatility and performance. In addition, the wide applications of pre-training models in cross-domains [24], [25], such as speech recognition and computer vision, demonstrate their advantages of utilising unlabeled data to help learn robust feature representations on limited labeled data. A comprehensive study on unsupervised pretraining [26] for Transformer-based speech recognition was conducted in paper [27], focusing on Masked Predictive Coding (MPC). Zhou et al. [28] presented a unified Vision-Language Pre-training (VLP) model, which was pre-trained on many image-text pairs.

In network traffic classification, Hendrycks et al. [29] showed that although pre-training may not improve performance on traditional classification metrics, it improves model robustness and uncertainty estimates. Through extensive experiments on label corruption, class imbalance, and more, they demonstrated large gains from pre-training and complementary effects with task-specific methods. The first attempt was made to provide a generative pre-trained model, NetGPT [30], for traffic understanding and generation tasks. Through multi-mode network traffic modeling, text input, header field cleaning, packet segmentation and label and prompt merging are unified, so as to optimize the adaptability of the pre-trained model to diverse tasks. Horowicz et al. [31] proposed using unsupervised Contrastive Learning (CL) [32], [33] to enhance flow image samples and alleviate the issue of small-sample flow classification. Lin et al. [12] proposes a new traffic feature representation model called ET-BERT, which extracts context packet-level representation from large-scale traffic raw packets to improve the accuracy of downstream classification tasks. Ferrag et al. [34] proposes a novel network-based cyber threat detection method using LLMs and introduces a privacy preserving encoding approach called fixed-length language encoding (FLLE). The paper implements and trains the BERT architecture from scratch for multi-category classification and adopts FalconLLM as an incident response and recovery system.

In conclusion, utilising LLM is a promising approach for achieving robust IDS in both in-vehicle and extra-vehicle scenarios. Therefore, this paper employs LLM to implement intrusion detection in vehicular networks, emphasising the capability of BERT to capture bidirectional contextual features and effectively generalise. This addresses the limitations present in existing ML/DL-based IDS [35], which may overlook potential contextual features.

III. Preliminary

A. Problem Formulation

1) Definitions Related to Network Traffic:

- Packet: In the extra-vehicle network, a packet represents the smallest unit of a flow. The in-vehicle network refers to CAN data frames, which consist of two main parts: CAN ID and Payload Data.
- Flow: Flow and session definitions are only for extra-vehicle networks. Flow identification is based on a quintuple consisting of the source address, the destination address, the source port, the destination port, and the TCP/UDP protocol. It is denoted as $F = p_1, p_2, \ldots, p_n$, where p_i represents packets with the same quintuple.
- Session: Session is also known as bi-directional flow when the same source address, destination address, source port, and destination port are used for up-flow and down-flow. It is notated as $S = F_{src \to dst} + F_{dst \to src}$.

2) Semantic List (SL): Semantic List (SL) is divided into BSL and USL, which are the outputs of BSE and USE, respectively. BSL is an unlabelled bi-directional semantic list consisting of two fields, byte sentence_a and byte sentence_b, which corresponds to the inputs text_a and text_b of the BERT. USL is a labelled uni-directional semantic list consisting of byte sentence_a and label. It can be seen that byte sentence is the basic unit of composing SL. Following vehicle network, byte sentences can be classified into in-vehicle network CAN Byte Sentences (CBS) and extra-vehicle network Traffic Byte Sentences (TBS).

B. BERT Model Architecture

BERT is a neural network model that relies entirely on an attention mechanism for parallel computation. Each bidirectional transformer encoder has the same structure, including a multi-head self-attention mechanism module, a feed-forward neural network module, and a residual connection and layer normalisation module. The transformer block used in this method is characterised by its one-to-one correspondence and the exact dimensions for inputs and outputs. The one-to-one correspondence in transformer enables it to process the positional information of each unit in the sequence through positional encoding. In addition, allowing the model to compute the feature representations of different positions in parallel can better handle long sequences and improve the training efficiency of the model compared to serial sequential processing such as LSTM.

1) Multiple Self-Attention Mechanism: The attention mechanism is a function that maps a query Q, and a set of key-value (K-V) pairs with M elements to a value. The existing Scaled Dot-Product Attention (SDPA) is defined as follows.

$$Attention_{SDPA}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where N and M denote the lengths of the query and key (or value); D_k and D_v denote the dimensions of the key (or query) and value; the scaling factor $\frac{1}{\sqrt{d_k}}$ is used for normalisation, decoupling the vector dimension from the softmax distribution, and avoiding gradient vanishing during training.

2) Feed-Forward Neural Network: After the self-attention mechanism, the bidirectional transformer encoder includes a feed-forward neural network. This network performs non-linear transformations on the input at each position to capture more complex features. The Feed-Forward Neural Network (FFN) consists of two fully connected layers. The first layer uses the activation function ReLU to enhance the model's non-linear expression ability. The corresponding FFN function is as follows:

$$FFN = \max\left(0, XW_1 + b_1\right) W_2 + b_2 \tag{2}$$

where X is the input; W_1 and b_1 are the learnable parameters of the first ReLU fully connected layer; W_2 and b_2 are the learnable parameters of the second fully connected layer.

3) Residual Connection and Layer Normalization: This part consists of two parts: residual connection and layer normalisation. In a residual connection, the input is added directly to the output of one layer as the input of the next layer. This approach is typically used for multi-layer network training and focuses the network only on the current difference. The input of each neural network layer is normalised using layer normalisation to speed up model convergence and enhance generalisation ability. $LayerNorm$ function is expressed as follows:

$$LayerNorm(X) = \alpha \odot \frac{X - \mu}{\sigma} + \beta \tag{3}$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 + \varepsilon} \tag{5}$$

where α and β are learnable parameter vectors; $\odot$ denotes element-wise multiplication; ε is a small constant added for numerical stability.

In the transformer block, both the multi-head attention mechanism and feed-forward neural network are followed by residual connection and layer normalisation, and the expression is as follows:

$$LayerNorm(X + MultiHeadAttention(X)) \tag{6}$$

$$LayerNorm(X + FeedForward(X)) \tag{7}$$

where X denotes the input of the multi-head attention mechanism or the feed-forward neural network.

IV. Proposed Framework: IoV-BERT-IDS

A. Framework

Fig. 2 visually represents the IoV-BERT-IDS framework, illustrating its four main modules: the semantic extractor, the input embedding, the IoV-BERT-IDS pre-training, and the IoV-BERT-IDS fine-tuning for learning a general traffic representation and performing network intrusion detection in the IoV scenario.

1) Semantic Extractor: As network traffic differs significantly from natural language in semantics, applying BERT directly to network intrusion detection is not feasible. To bridge this gap, we propose a Semantic Extractor module that transforms network traffic data into a semantic list. The module primarily consists of Bidirectional SE (BSE) and Unidirectional SE (USE), respectively employed during the pre-training and fine-tuning stages. The present approach involves exclusively utilising extra-vehicle network data for the pre-training phase. Subsequently, in the fine-tuning phase, regardless of whether the data stems from in-vehicle or extra-vehicle network traffic, packets are processed using the USE to derive the USL, which is then fed as input to the model.

2) Input Embedding: Input embedding involves combining token, segment, and position embeddings to form a comprehensive numerical representation of the input sequence. This process transforms the byte sentence into a format that can be effectively processed by the IoV-BERT-IDS, preserving crucial semantic, syntactic, and positional information necessary for the model to learn.

3) IoV-BERT-IDS Pre-Training: Given the unlabeled BSL, the pre-trained model is required to learn a general representation of traffic, completing the basic level of education during the pre-training stage. To learn traffic representations with relative semanticisation, two tasks are proposed as part of the pre-training phase: Masked Byte Word Model and Next Byte Sentence Prediction.

4) IoV-BERT-IDS Fine-Tuning: By providing labelled USL to the pre-trained model during the fine-tuning stage, intrusion detection for traffic at different vehicle networks can be achieved. As input for fine-tuning the model, corresponding labelled data is all that is required when addressing various downstream tasks.

B. Semantic Extractor

To ensure that the pre-trained model achieves a meaningful representation of traffic in the pre-training phase, SE considers two aspects: First, how should the data be partitioned to have the sequence's contextual semantics? The analogy here can be drawn to how a natural language without breaks should be divided into chunks of words to give it reasonably fluent semantics. Since pcap files usually split datagrams by double bytes, these two bytes can be regarded as one unit, and the two units form one word, thus transforming the raw traffic data into byte sentences by the USE. Second, how is the contextual relationship between traffic sequences represented? For BSE, successive packets in the same session can be analogised to the contextual relationships in an NLP paragraph.

1) Unidirectional Semantic Extractor: It works by breaking the hexadecimal string into the smallest units, called byte units, according to every two characters and then merging BU into a word called Byte Word (BW). After the hexadecimal string is processed, it is transformed into a BS made up of BW. USE converts the CAN data into CBS, while the extra-vehicle network traffic data is converted into TBS.

During the fine-tuning stage, the USE takes in labelled traffic data as input and produces USL as output. USL comprises two components: labels and byte sentence_a. For extra-vehicle network traffic data, the payload of each packet is extracted and

processed through USE to derive TBS, representing the value of the byte sentence_a field. As for in-vehicle network CAN data, each CAN data point's CAN ID and DATA[0-7] (refer to Fig. 5 for the CAN data structure) undergo USE transformation to yield CBS.

2) Bidirectional Semantic Extractor: The BSE builds upon the USE by incorporating a sliding window mechanism. It selects consecutive packets within a session using a sliding window of length 2 and step size 1, where each pair of adjacent packets serves as model inputs byte sentence_a and byte sentence_b in Fig. 2.

During the pre-training phase, BSE ingests unlabeled network traffic data as its input and yields BSL as its output. The BSE algorithm operates as Algorithm 1. It iterates through the packets within PCAP files, extracting fundamental details such as timestamps, source and destination IP addresses, ports, protocols, and packet payloads. Subsequently, each session (identified by a unique 5-tuple) employs a sliding window mechanism to select adjacent payloads. These selected payloads are then fed into the USE function to generate byte sentence_a and byte sentence_b, culminating in the production of BSL. The USE function in this context embodies the functionality of the Unidirectional Semantic Extractor, tasked with transforming packet messages into byte sentences. Initially, the packet messages are segmented into pairs of characters, and subsequently, consecutive pairs of characters are systematically assembled into byte sentences.

Algorithm 1: Bidirectional Semantic Extractor.

Input: PCAP files
Output: BSL

Initialize BSL;
for each PCAP **do**
  Initialize $Sessions$;
  for each packet in PCAP **do**
    $ID \leftarrow$ [srcip, dstip, srcport, dstport, protocol]
    $pl \leftarrow$ hex(payload)[8:]
    if ID exists in $Sessions$ **then**
      $Sessions[ID]$.append(pl)
    else
      $Sessions[ID] \leftarrow [pl]$
    end if
  end for
  for each $session$ in $Sessions$ **do**
    $s\_payload \leftarrow$ the number of packets in the session
    for packet index i in $session$ **do**
      if $i \neq (s\_payload - 1)$ **then**
        // It's not the last pair of adjacent packets.
        $bs\_a \leftarrow USE(session[i])$
        $bs\_b \leftarrow USE(session[i+1])$
      else if $s\_payload == 1$ **then**
        // There's only one packet in the session.
        $mid \leftarrow$ length of session[i] / 2
        $bs\_a \leftarrow USE(session[i][0 : mid])$
        $bs\_b \leftarrow USE(session[i][mid :])$
      else
        continue
      end if
      BSL.append([bs_a, bs_b])
    end for
  end for
end for

C. Input Embedding

The input sequence uses the sum of three embedding vectors to construct the complete token representation: token embedding, position embedding, and paragraph embedding.

- *Token embedding:* The payload is converted into fixed-dimensional vectors. Before this, the input data is first processed by looking up the dictionary to learn the token representation.
- *Segment embedding:* It is closely related to the token [SEP]. To represent the upper and lower sentences, the embedding uses only 0 and 1 based on the context sentences segmented by [SEP]. Segment embeddings are all-0 if there is only one sentence as input.
- *Position embedding:* BERT cannot encode the orderliness of the input sequence since traffic data transmission is closely related to the order. Thus, positional embedding is needed to allow the model to learn sequential information from sequences based on relative positions.

In this paper, we use BPE to represent tokens. Each token unit ranges from 0 to 65535, and the dictionary size $|V|$ is up to 65536. In addition, four unique tokens, [CLS], [SEP], [PAD], and [MASK], are added. The first token of each sequence is always [CLS]. The token [PAD] is a padding symbol to satisfy the minimum length requirement. Given that all byte pairs can be enumerated, Out-of-Vocabulary (OOV) words may not occur during token embedding. Nevertheless, a unique token [UNK] is introduced to address the possibility of OOV occurrences to represent words not present in the dictionary. Sentence pairs are separated using [SEP] to represent different segments. The token [MASK] is used in the masked language modeling task to learn the flow context during pre-training.

D. IoV-BERT-IDS Pre-Training

1) Masked Byte Word Model: The MBWM task is similar to BERT's MLM, the only difference being that the MBWM task is pre-trained for BW with relative semanticisation for capturing dependencies between BWs. This design allows each BW prediction to focus on the context of the entire BS, enabling deep bidirectional coding capabilities. During pre-training, each token in the input sequence is randomly masked with a 15% probability. As shown in Fig. 3, to be closer to the subsequent downstream task fine-tuning, the following operations are performed for these 15% tokens:

- With an 80% probability, the input corresponding to the token is replaced by a [MASK] token.
- With a 10% probability, another token randomly replaces the input corresponding to the token.
- With a 10% probability, the input corresponding to the token remains unchanged.

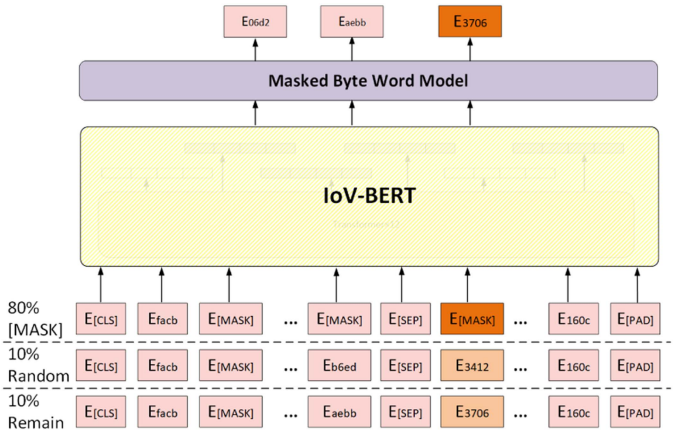


Fig. 3. The schematic diagram of the MBWM pre-training task.

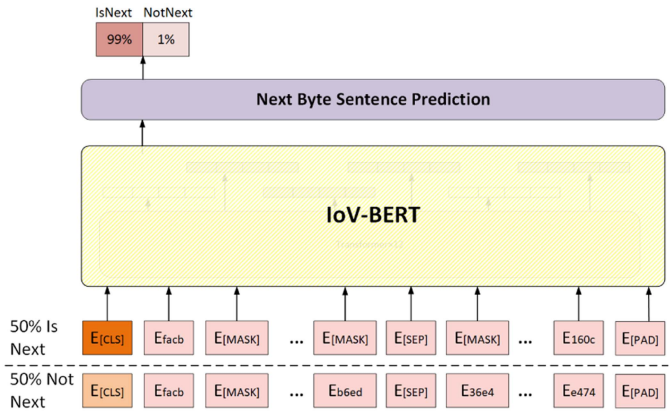


Fig. 4. The schematic diagram of the NBSP pre-training task.

Simply put, the MBWM task predicts the token at the masked position based on the context. The cross-entropy loss L_{MBWM} of the MBWM task is defined as follows:

$$L_{MBWM} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} y_{jc}\log\left(P\left(token_c|MASK_j\right)\right) \quad (8)$$

where k is the total number of tokens masked in the input, $MASK_j$ denotes the j-th masked token, $token_c$ denotes the c-th token in the dictionary, and n is the batch_size. $P(token_c|MASK_j)$ denotes the probability that the j-th masked token is predicted as the c-th token in the dictionary, and y_{jc} is a sign function, which is 1 if [MASK] is predicted correctly, and 0 otherwise.

2) Next Byte Sentence Prediction: Similar to BERT's NBSP, the NBSP in this paper is also a binary classification task. For the model to understand the dependency between two BSs, we use a 50% probability to randomly replace the sentence in the input, as shown in Fig. 4. Therefore, for the NBSP task, the pre-trained model needs to make a binary input prediction to determine whether it is continuous. The cross-entropy loss function here is denoted as L_{NBSP} and defined as follows:

$$L_{NBSP} = -\frac{1}{n}\sum_{i=1}^{n} y_{ic}\log\left(P\left(y_c \mid p_i\right)\right), c \in \{0,1\} \quad (9)$$

where p_i denotes the i-th sentence pair, y_c indicates whether it belongs to a continuous pair, c = 0 means it is not a continuous sentence, and $P(y_c \mid p_i)$ represents the probability of the i-th sentence pair being predicted as class c.

In summary, the final pre-training objective is the sum of the two losses mentioned above, which is defined as:

$$L = L_{MBWM} + L_{NBSP} \quad (10)$$

E. IoV-BERT-IDS Fine-Tuning

By exchanging appropriate inputs and outputs, the Transformer allows BERT to adapt to many downstream tasks, whether individual or pairs of texts. Thus, domain fine-tuning for intrusion detection tasks only requires modifying the inputs and outputs of the end-to-end model. For fine-tuning, the [CLS] corresponding output from the last transformer encoder block is fed into the multi-classifier for prediction. To achieve multiclassification, we use the simplest fully connected layer followed by the softmax layer. Fig. 2 illustrates how fine-tuning is similar to pre-trained NBSP, where only the binary classifier of the NBSP task needs to be replaced with a multi-classifier, and the data preprocessing method must match the downstream task. In contrast to pre-trained data, fine-tuning uses labelled data and pre-processes datasets based on the downstream task. An in-vehicle network intrusion detection fine-tuning dataset can be created using raw CAN data preprocessed into CBS. Whether an in-vehicle or extra-vehicle network, the payload is essentially a hexadecimal string with a similar token representation after semantic transformation by SE. So, the IoV-BERT-IDS here can simultaneously adapt to in-vehicle and extra-vehicle network intrusion detection.

V. Experimental Setup

A. Datasets

1) Pre-Training Dataset: The pre-training dataset for this study consists of approximately 40 G of PCAP files. These files were collected using Wireshark software, leveraging router port mirroring functionality. This functionality duplicates network traffic from the router to the PC, allowing Wireshark to capture and analyse the replicated data. The captured traffic encompasses various communication activities from numerous mobile devices connected to the router. Including TCP, UDP, ICMP, and other protocols ensures that the model, during its pre-training phase, can effectively learn and capture features across varied network communication scenarios.

2) Extra-Vehicle Network Datasets: Due to the temporary unavailability of public datasets specifically designed for vehicular network traffic, this paper employs different network traffic intrusion detection datasets to assess the prediction performance of IoV-BERT-IDS in the vehicular network scenario. Given that the method outlined in this paper requires payload data from the original traffic, two datasets, CICIDS and BoT-IoT, have been selected for experimentation. By comparing the five-tuple of each packet in the CSV file and the PCAP file, we re-divided the packets in the PCAP file. The following provides an introduction to the two datasets.

TABLE I
CICIDS DATASET

Classname	CSV records		pcap flows	
	Original	Exp 1	Original	Exp 1
BENIGN	2,273,097	20,000	92,460	10,000
Hulk	231,073	20,000	14,108	10,000
PortScan	158,930	20,000	158,492	10,000
DDoS	128,027	20,000	24,091	10,000
GoldenEye	10,293	10,293	5,319	5,319
FTP-Patator	7,938	5,796	648	648
SSH-Patator	5,897	5,499	660	660
Slowloris	5,796	7,938	3,845	3,845
Slowhttptest	5,499	5,897	4,071	4,071
Bot	1,966	1,966	1,204	1,204
Brute Force	1,507	1,507	1,274	1,274
XSS	652	652	625	625
Infiltration	36	/	5	/
Sql Injection	21	/	3	/
Heartbleed	11	/	1	/

TABLE II
BOT-IOT DATASET

Classname	CSV records		pcap flows	
	Original	Exp 1	Original	Exp 1
DDoS	1,926,624	10,000	552,840	10,000
DoS	1,650,260	10,000	527,573	10,000
Reconnaissance	91,082	10,000	44,053	10,000
Normal	477	477	1,150	1,150
Theft	79	79	782	782

Fig. 5. Feature structure of CAN packets.

CICIDS: We selected twelve attack types of PCAP files, which are used for the fine-tuning stage of IoV-BERT-IDS in this paper. Table I shows detailed information about the subset of CICIDS for experiments.

BoT-IoT: The captured PCAP files are 69.3 G large, with over 72,000,000 records. We only select five per cent of the full BoT-IoT datasets and detailed information is shown in Table II.

3) In-Vehicle Network Datasets: As shown in Fig. 5, CAN dataset includes various fields:

- Timestamp represents the recorded time in seconds;
- CAN ID signifies the identifier of the CAN message in hexadecimal format (e.g., 043f);
- DLC indicates the number of data bytes ranging from 0 to 8;
- DATA[0-7] encompasses the byte-wise data values;
- Label is the class name of CAN packet.

Car-Hacking Dataset: Datasets were constructed by logging CAN traffic via the OBD-II port from a real vehicle while message injection attacks were performed, which include DoS attack, fuzzy attack, spoofing the drive gear, and spoofing the RPM gauge. The details of this dataset are shown in the Table III.

In-vehicle network intrusion detection challenge [36]: For fine-tuning purposes, this paper selects a preliminary dataset that encompasses three vehicle types (CHEVROLET Spark, HYUNDAI Sonata, KIA Soul) and various attack types (Flooding, Fuzzy, Malfunction). Additionally, the training dataset includes a Class field for each record. For reference, this dataset is abbreviated IVN-IDS. Table IV shows the details of this dataset.

TABLE III
CAR-HACKING DATASET

Classname	Original	Exp 2
Normal	23,497,859	10,000
RPM	1,803,195	10,000
Gear	597,252	10,000
DoS	587,521	10,000
Fuzzy	491,847	10,000

TABLE IV
IN-VEHICLE NETWORK INTRUSION DETECTION CHALLENGE(IVN-IDS DATASET

Classname	Original	Exp 2	Vehicle	Original	Exp 3
Attack Free	1,172,729	10,000	CHEVROLET Spark	291893	/
			HYUNDAI Sonata	353,779	8,000
			KIA Soul	527,057	2,000
Flooding	48,164	10,000	CHEVROLET Spark	14,999	/
			HYUNDAI Sonata	17,093	8,000
			KIA Soul	16,072	2,000
Fuzzy	33,751	10,000	CHEVROLET Spark	3043	/
			HYUNDAI Sonata	9,095	8,000
			KIA Soul	21,613	2,000
Malfunction	16,967	10,000	CHEVROLET Spark	3995	/
			HYUNDAI Sonata	8,202	8,000
			KIA Soul	4,770	2,000

TABLE V
EQUIPMENT CONFIGURATION

Name	Specification
GPU	NVIDIA GeForce RTX 4080
GPU Memory	16Gb
Python Version	3.9.0
CUDA Version	12.2
Pytorch Version	2.0.1

TABLE VI
EXP 1: EVALUATION METRIC COMPARISON

Model	CICIDS				BoT-IoT			
	Precision	Recall	F1	ACC	Precision	Recall	F1	ACC
AE	0.86	0.84	0.84	0.94	0.89	0.94	0.91	0.94
VAE	0.86	0.89	0.87	0.97	0.94	0.95	0.95	0.96
ByteSGAN	0.89	0.87	0.85	0.93	0.94	0.97	0.95	0.97
IoV-BERT-IDS	**0.99**	**1.00**	**1.00**	**0.99**	**1.00**	**1.00**	**1.00**	**1.00**

The bold values represent the best performance of this indicator in each column to highlight the superior performance of our proposed method.

B. Experimental Environment

The experimental environment is shown in Table V, the GPU is NVIDIA GeForce RTX 4080, the GPU Memory is 16 GB, the version of Python used is 3.9.0, and the model building is mainly using Pytorch, the version is 2.0.1.

C. Evaluation Metrics

The performance evaluation indicators in this article include Precision, Recall, F1-score, and Accuracy derived from four parameter indicators TP, TN, FP, and FN. The following is a detailed description of this article's performance evaluation indicators.

1) Precision, also known as positive predictive value or precision rate, refers to the proportion of true positive samples

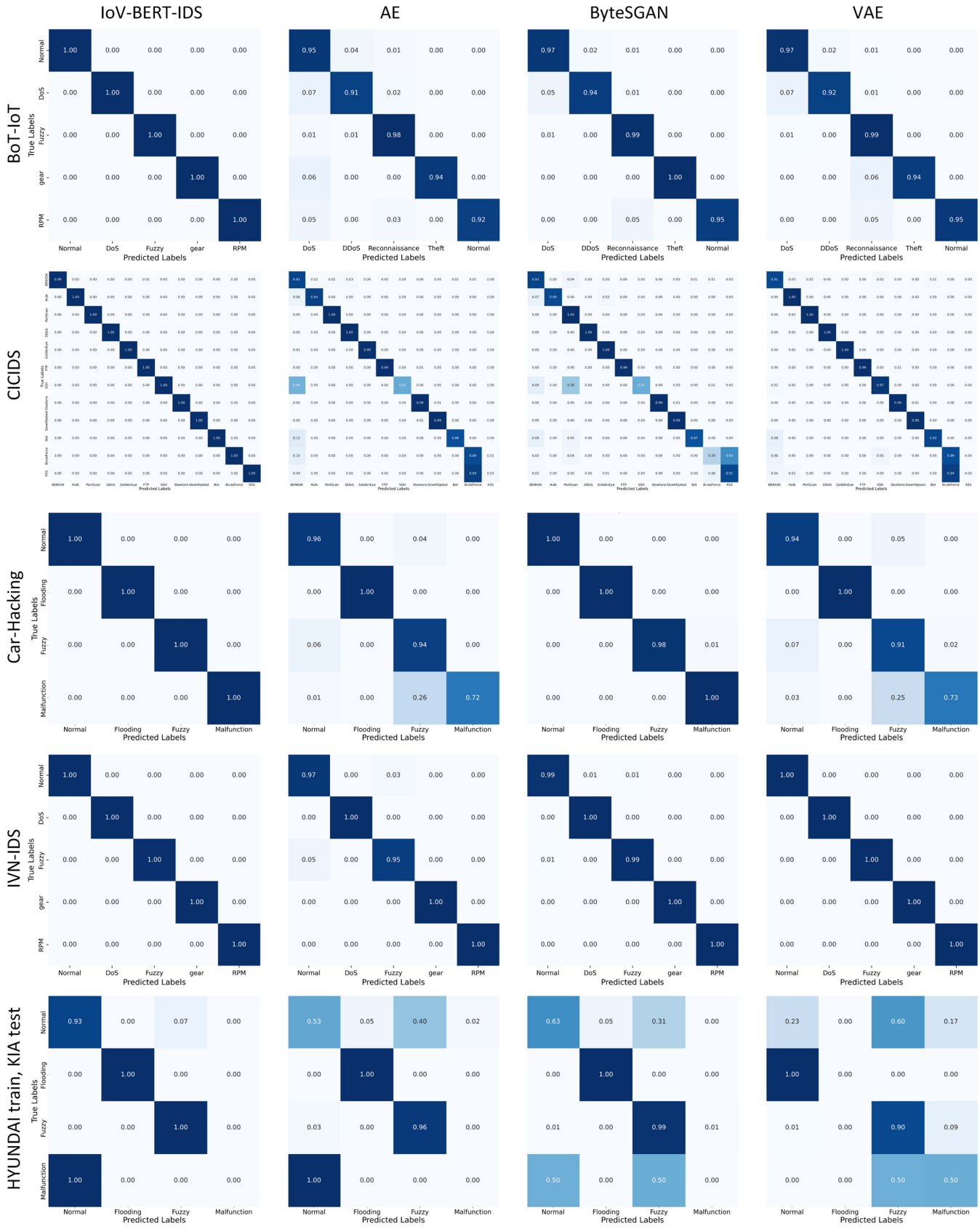


Fig. 6. Confusion matrices of classification performance.

among the samples that we predicted as positive. The formula is (11), where TP is the number of true positives and FP is the number of false positives.

$$Precision = \frac{TP}{TP + FN} \tag{11}$$

2) Recall, also known as sensitivity or true positive rate, refers to the proportion of positive samples that we predicted as positive among the samples that are actually positive. The formula is (12), where TP is the number of true positives and FN is the number of false negatives.

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

3) F1-score, also known as F1 value or F1 measure, is a weighted average of precision and recall. It can reflect the model's ability to identify and distinguish positive samples. The formula is (13).

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

4) Acc, also known as accuracy or correct rate, refers to the proportion of correctly classified samples among the total number of samples. The formula is (14), where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

VI. Experimental Results

In this paper, the IoV-BERT-IDS method follows a 'pre-training + fine-tuning' approach. Consequently, we have chosen ByteSGAN [9], VAE [37], and AE [38] as the comparative methods. Unlike our approach, the first three methods rely on extracted flow-level features, whereas IoV-BERT-IDS directly uses raw traffic data.

The confusion matrices for these experiments are depicted in Fig. 6. In conclusion, the experiments demonstrate IoV-BERT-IDS's superior classification performance on IoV. Its robust representation capabilities, generalisation, and adaptability contribute to its effectiveness in handling complex intrusion detection tasks.

A. Performance Comparison of IDS in Extra-Vehicle Networks (Experiment 1)

In an effort to comprehensively assess the predictive capabilities of IoV-BERT-IDS when deployed within an extra-vehicle network, a series of meticulous experiments were carried out utilizing the well-established CICIDS and BoT-IoT datasets.

Table VI demonstrates that on the CICIDS dataset, the IoV-BERT-IDS model outperformed others, achieving the highest scores across all metrics, with precision, recall, F1-score, and accuracy values of 0.99, 1.00, 1.00, and 0.99, respectively. In contrast, the other three models lagged slightly behind. Furthermore, on the BoT-IoT dataset, the IoV-BERT-IDS model once again demonstrated outstanding performance.

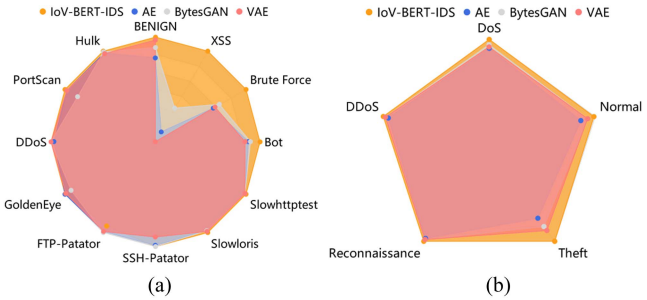


Fig. 7. Classification precision comparison. (a) CICIDS. (b) BoT-IoT.

To provide a more intuitive and holistic understanding of these results, Fig. 7 presents two radar charts, each axis labelled with tags representing different types of network traffic attacks. These charts visually depict the classification accuracy achieved by IoV-BERT-IDS, AE, ByteSGAN, and VAE across both datasets. Notably, IoV-BERT-IDS is represented by a distinctive red line, covering a larger area on the radar charts. This visual representation signifies that the IoV-BERT-IDS model exhibits higher precision in classifying various types of traffic attacks in the extra-vehicle network environment. This indicates that IoV-BERT-IDS outperforms other models in accurately identifying and categorising various network attacks, underscoring its effectiveness in enhancing cybersecurity measures in such environments.

B. Performance Comparison of IDS in In-Vehicle Networks (Experiment 2)

To delve deeper into assessing the classification performance of the BERT model in the in-vehicle network intrusion detection domain, comprehensive experiments were conducted utilising the Car-Hacking and IVN-IDS datasets. These experiments encompassed comparisons between multiple models, including ByteSGAN, VAE, AE, and IoV-BERT-IDS, utilising solely CAN ID and payload data for training and testing.

Table VII summarises the classification metrics derived from these comparative experiments, including precision, recall, F1-score, and accuracy. It's evident from the table that on the IVN-IDS dataset, the IoV-BERT-IDS model achieved the highest scores across all metrics, with precision, recall, F1-score, and accuracy values of 0.9996. Similarly, on the Car-Hacking dataset, the IoV-BERT-IDS model maintained its superiority in precision, recall, and F1-score, with values of 0.9998, 0.9998, and 0.9997, respectively. However, ByteSGAN edged slightly ahead in accuracy with a score of 0.9999. Overall, the IoV-BERT-IDS model exhibited outstanding performance on both datasets, displaying high precision, recall, and F1-scores. Furthermore, Fig. 8 visually represents the precision, recall, and F1-score on the IVN-IDS dataset, emphasising the exceptional performance of the IoV-BERT-IDS model. While the ByteSGAN model demonstrated commendable results on the Car-Hacking dataset, the IoV-BERT-IDS model slightly outperformed it in recall and overall accuracy. The significant precision and recall values achieved by the IoV-BERT-IDS model imply its ability to distinguish between normal operational behaviour and various

TABLE VII
EXP 2: EVALUATION METRIC COMPARISON

Model	IVN-IDS				Car-Hacking			
	Precision	Recall	F1	ACC	Precision	Recall	F1	ACC
AE	0.9271	0.9049	0.9085	0.9122	0.9949	0.9949	0.9949	0.9949
VAE	0.9141	0.8956	0.8989	0.9022	0.9835	0.9835	0.9834	0.9835
ByteSGAN	0.9949	0.9954	0.9951	0.9953	**0.9989**	0.9989	**0.9988**	0.9989
IoV-BERT-IDS	**0.9996**	**0.9996**	**0.9996**	**0.9996**	0.9971	**0.9998**	0.9985	**0.9997**

The bold values represent the best performance of this indicator in each column to highlight the superior performance of our proposed method.

TABLE VIII
EXP 3: EVALUATION METRIC COMPARISON

IVN_Exp3	Normal			Flooding			Fuzzy			Malfunction		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
AE	0.34	0.53	0.41	0.95	1.00	0.98	0.71	0.96	0.81	0.00	0.00	0.00
VAE	0.18	0.23	0.20	0.00	0.00	0.00	0.45	0.90	0.60	**0.66**	**0.50**	**0.57**
ByteSGAN	**0.56**	0.63	0.59	0.95	1.00	0.97	0.55	0.99	0.70	0.00	0.00	0.00
IoV-BERT-IDS	0.48	**0.93**	**0.63**	**1.00**	**1.00**	**1.00**	**0.93**	**1.00**	**0.96**	0.00	0.00	0.00

The bold values represent the best performance of this indicator in each column to highlight the superior performance of our proposed method.

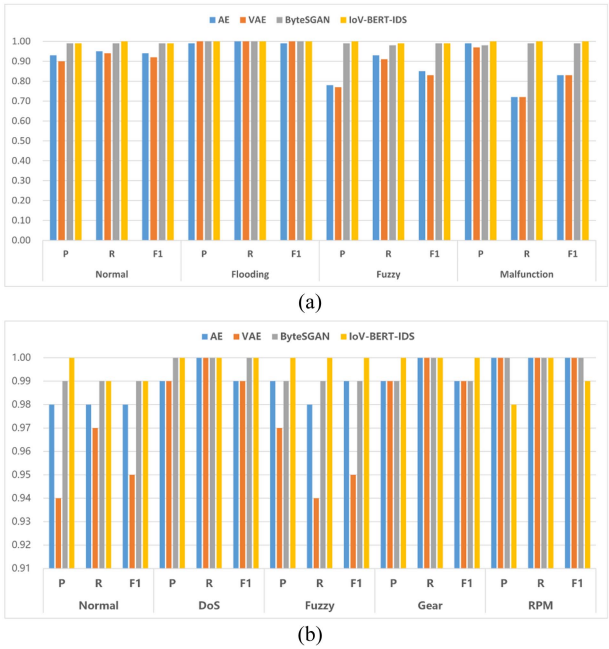


Fig. 8. Classification performance comparison: Precision, Recall, F1-score. (a) IVN-IDS. (b) Car-Hacking.

attack scenarios while maintaining low rates of false positives and false negatives. Such reliability is crucial when deploying intrusion detection systems in vehicular environments, where the consequences of security vulnerabilities can be severe.

C. Generalization Comparison of IDS in In-Vehicle Networks (Experiment 3)

To further assess the extent of improvement in IoV-BERT-IDS's model generalisation performance, a specific experiment was conducted utilising data from the IVN-IDS dataset. Specifically, data from HYUNDAI Sonata was chosen for training, while data from KIA Soul was used for testing, aiming to evaluate the model's ability to generalise across different vehicle models.

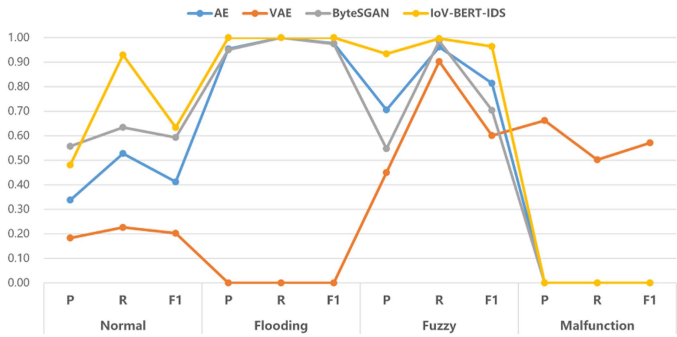


Fig. 9. Classification generalization in in-vehicle networks.

Table VIII provides a detailed overview of the classification metrics derived from this comparative experiment. The IoV-BERT-IDS model exhibited excellent performance on the IVN_Exp3 dataset, particularly in identifying Flooding and Fuzzy attacks, with high precision, recall, and F1-scores. However, all models struggled to effectively identify Malfunction attacks, possibly due to Malfunction alerts resembling normal traffic.Additionally, Fig. 9 presents the performance of each model, highlighting the significant improvement achieved by IoV-BERT-IDS in terms of model generalisation between different vehicle models (KIA Soul and HYUNDAI Sonata) compared to other models. The results demonstrate a significant enhancement in IoV-BERT-IDS's model generalisation performance, indicating its superior capability in handling data from diverse vehicle models. This suggests that IoV-BERT-IDS exhibits a higher level of adaptability to the CAN protocol and potential differences in encoding rules between vehicles. Consequently, IoV-BERT-IDS demonstrates stronger generalisation performance when detecting intrusions and anomalies in in-vehicle networks across a broader range of vehicle models.

VII. CONCLUSION

With the rapid evolution of intelligent connected vehicles, ensuring the security of vehicular networks has become a pressing concern [39]. This paper addresses the limitations of traditional

security models in handling the complex and diverse connections within the Internet of vehicles. The proposed framework, IoV-BERT-IDS, leverages BERT to acquire a universal representation of raw traffic data in IoV. This approach enhances accuracy through unsupervised pre-training and fine-tuning with minimal data. Addressing the challenge of ambiguous semantics in traffic data, the semantic extractor transforms raw information into contextual semantic traffic pairs. This comprehensive approach ensures the model has a robust understanding of the contextual semantics inherent in IoV traffic data. Furthermore, the introduction of two pre-training tasks enriches IoV-BERT-IDS with bidirectional contextual features, contributing significantly to learning and capturing contextual patterns and features. The effectiveness of IoV-BERT-IDS is demonstrated through thorough validation of diverse datasets, including CICIDS, BoT-IoT, Car-Hacking, and IVN-IDS. Given the challenges of limited resources in vehicles and roadsides, future research can explore more efficient and lightweight BERT models [40]. Optimising model structures, parameters, and training algorithms to adapt to computational resource constraints in vehicular network environments.

REFERENCES

[1] X. Zhou et al., "Spatial–temporal federated transfer learning with multi-sensor data fusion for cooperative positioning," *Inf. Fusion*, vol. 105, 2024, Art. no. 102182. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253523004980

[2] S. Sharma and B. Kaushik, "A survey on Internet of Vehicles: Applications, security issues & solutions," *Veh. Commun.*, vol. 20, 2019, Art. no. 100182.

[3] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[4] Z. Du et al., "Integrated sensing and communications for V2I networks: Dynamic predictive beamforming for extended vehicle targets," *IEEE Trans. Wireless Commun.*, vol. 22, no. 6, pp. 3612–3627, Jun. 2023.

[5] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures: Challenges and future directions," *IEEE Netw.*, vol. 31, no. 5, pp. 50–58, Sep. 2017.

[6] X. Zhou et al., "Decentralized P2P federated learning for privacy-preserving and resilient mobile robotic systems," *IEEE Wireless Commun.*, vol. 30, no. 2, pp. 82–89, Apr. 2023.

[7] X. Li, Z. Hu, M. Xu, Y. Wang, and J. Ma, "Transfer learning based intrusion detection scheme for Internet of Vehicles," *Inf. Sci.*, vol. 547, pp. 119–135, 2021.

[8] Y. Lee, Y.-E. Kim, J.-G. Chung, and S. Woo, "Real time perfect bit modification attack on in-vehicle CAN," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 15154–15171, Dec. 2023.

[9] P. Wang, Z. Wang, F. Ye, and X. Chen, "BytesGAN: A semi-supervised generative adversarial network for encrypted traffic classification in SDN edge gateway," *Comput. Netw.*, vol. 200, 2021, Art. no. 108535.

[10] T.-L. Huoh, Y. Luo, P. Li, and T. Zhang, "Flow-based encrypted network traffic classification with graph neural networks," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1224–1237, Jun. 2023.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[12] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, "ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification," in *Proc. ACM Web Conf.*, New York, NY, USA, 2022, pp. 633–642.

[13] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, vol. 1, pp. 108–116.

[14] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework," *Future Gener. Comput. Syst.*, vol. 110, pp. 91–106, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X19325105

[15] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Veh. Commun.*, vol. 21, 2020, Art. no. 100198.

[16] X. Zhou et al., "Federated distillation and blockchain empowered secure knowledge sharing for Internet of Medical Things," *Inf. Sci.*, vol. 662, 2024, Art. no. 120217.

[17] L. Yang and A. Shami, "A transfer learning and optimized CNN based intrusion detection system for Internet of Vehicles," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 2774–2779.

[18] T. Alladi, V. Kohli, V. Chamola, F. R. Yu, and M. Guizani, "Artificial intelligence (AI)-empowered intrusion detection architecture for the Internet of Vehicles," *IEEE Wireless Commun.*, vol. 28, no. 3, pp. 144–149, Jun. 2021.

[19] L. Nie, Z. Ning, X. Wang, X. Hu, J. Cheng, and Y. Li, "Data-driven intrusion detection for intelligent Internet of Vehicles: A deep convolutional neural network-based method," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2219–2230, Oct.–Dec. 2020.

[20] T. Yu, G. Hua, H. Wang, J. Yang, and J. Hu, "Federated-LSTM based network intrusion detection method for intelligent connected vehicles," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 4324–4329.

[21] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A multitiered hybrid intrusion detection system for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 616–632, Jan. 2022.

[22] N. Alkhatib, M. Mushtaq, H. Ghauch, and J.-L. Danger, "Can-BERT do it? Controller area network intrusion detection system based on BERT language model," in *Proc. IEEE/ACS 19th Int. Conf. Comput. Syst. Appl.*, 2022, pp. 1–8.

[23] L. Dong et al., "Unified language model pre-training for natural language understanding and generation," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 13063–13075.

[24] P. Wang, F. Ye, X. Chen, and Y. Qian, "DataNet: Deep learning based encrypted network traffic classification in SDN home gateway," *IEEE Access*, vol. 6, pp. 55380–55391, 2018.

[25] X. Zhou et al., "Hierarchical federated learning with social context clustering-based participant selection for Internet of Medical Things applications," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 4, pp. 1742–1751, Aug. 2023.

[26] Z. Wang, Z. Li, M. Fu, Y. Ye, and P. Wang, "Network traffic classification based on federated semi-supervised learning," *J. Syst. Archit.*, vol. 149, 2024, Art. no. 103091.

[27] D. Jiang et al., "A further study of unsupervised pretraining for transformer based speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 6538–6542.

[28] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and VQA," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, pp. 13041–13049.

[29] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, vol. 97, pp. 2712–2721.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre- training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[31] E. Horowicz, T. Shapira, and Y. Shavitt, "A few shots traffic classification with mini-flowpic augmentations," in *Proc. 22nd ACM Internet Meas. Conf.*, 2022, pp. 647–654.

[32] X. Zhou et al., "Personalized federation learning with model-contrastive learning for multi-modal user modeling in human-centric metaverse," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 4, pp. 817–831, Apr. 2024.

[33] X. Lin et al., "CETP: A novel semi-supervised framework based on contrastive pre-training for imbalanced encrypted traffic classification," *Comput. Secur.*, 2024, Art. no. 103892.

[34] M. A. Ferrag et al., "Revolutionizing cyber threat detection with large language models: A privacy-preserving bert-based lightweight model for iot/iiot devices," *IEEE Access*, vol. 12, pp. 23733–23750, Feb. 2024.

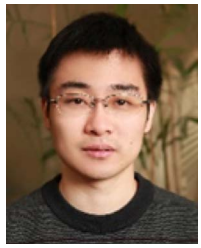
[35] X. Zhou et al., "Information theoretic learning-enhanced dual-generative adversarial networks with causal representation for robust OOD generalization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 17, 2023, doi: 10.1109/TNNLS.2023.3330864.
[36] M. L. Han, B. I. Kwak, and H. K. Kim, "Anomaly intrusion detection method for vehicular networks based on survival analysis," *Veh. Commun.*, vol. 14, pp. 52–63, 2018.
[37] C. Liu, R. Antypenko, I. Sushko, and O. Zakharchenko, "Intrusion detection system after data augmentation schemes based on the VAE and CVAE," *IEEE Trans. Rel.*, vol. 71, no. 2, pp. 1000–1010, Jun. 2022.
[38] G. D'Angelo and F. Palmieri, "Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial–temporal features extraction," *J. Netw. Comput. Appl.*, vol. 173, 2021, Art. no. 102890.
[39] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4298–4311, Apr. 2020.
[40] X. Zhou et al., "Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3191–3211, Oct. 2023.

Mengyi Fu (Graduate Student Member, IEEE) received the B.Sc. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2022. She is currently working toward the Ph.D. degree with the Nanjing University of Posts and Telecommunications, Nanjing, China. Her research interests include encrypted traffic identification, deep learning, and traffic prediction.

Pan Wang (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical & computer engineering from the Nanjing University of Posts &Telecommunications, Nanjing, China, in 2001, 2004, and 2013, respectively. He is currently a Full Professor with Nanjing University of Posts & Telecommunications. From 2017 to 2018, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Dayton, Dayton, OH, USA. He was a TPC member of IEEE CyberSciTech Congress. His research interests include AI-powered networking and security in B5G/6 G/IoT/Smart Grid/CFN, and AI-enabled Big Data analysis. He is also a reviewer for several journals, including IEEE TRANSACTION ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTION ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, IEEE INTERNET OF THINGS JOURNAL, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE ACCESS, *Computer Networks, Computer & Security*, *Computer Communications*, *Engineering Applications of Artificial Intelligence*, *Big Data Research*.

Minyao Liu received the bachelor's degree in management in 2022 from the Nanjing University of Posts and Telecommunications, where she is currently working toward the master's degree. Her research interests include traffic identification, deep learning, and anomaly detection.

Ze Zhang was born in Zhenjiang, Jiangsu, China, in 2000. He received the bachelor's degree in network engineering in 2022. He is currently working toward the master's degree in information network with the Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include computer networks, network security, anomaly detection and analysis.



Xiaokang Zhou (Member, IEEE) received the Ph.D. degree in human sciences from Waseda University, Tokyo, Japan, in 2014. He is currently an Associate Professor with the Faculty of Business Data Science, Kansai University, Suita, Japan. From 2012 to 2015, he was a Research Associate with the Faculty of Human Sciences, Waseda University. From 2016 to 2024, he was a Lecturer/Associate Professor with the Faculty of Data Science, Shiga University, Hikone, Japan. He has been a Visiting Researcher with the RIKEN Center for Advanced Intelligence Project (AIP), RIKEN, Japan, since 2017. He has been engaged in interdisciplinary research works in the fields of computer science and engineering, information systems, and social and human informatics. His research interests include ubiquitous computing, Big Data, machine learning, behavior and cognitive informatics, cyber-physical-social systems, and cyber intelligence and security. Dr. Zhou is a Member of the IEEE CS, and ACM, USA, IPSJ, and JSAI, Japan, and CCF, China.