A Survey Paper on Multi-Head CNN-BiLSTM Framework for Secure Intrusion Detection in Internet of Vehicles Networks

Mr. M. RAMESH [1], S. GOKILA [2]
[1]*Assistant professor, Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Tamilnadu, India*
[2]*PG Scholar, Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Tamilnadu, India*

***Abstract*—The Internet of Vehicles (IoV) links vehicles, roadside units and cloud services, so attacks can reach both the in-vehicle Controller Area Network (CAN) and the external communication links. Deep learning intrusion detection systems (IDS) are a common defence, but a single architecture sees only part of the problem: convolutional networks capture local patterns in a frame or flow, while recurrent networks capture order over time. Hybrid CNN-LSTM designs combine the two, and a multi-head CNN feeding a bidirectional LSTM (BiLSTM) extends this idea by extracting features at several scales in parallel and reading the resulting sequence in both directions. This paper surveys deep learning IDS for vehicular networks, with emphasis on CNN, recurrent and hybrid models, compares their datasets, architectures and reported performance, and identifies open problems: near-saturated accuracy on CAN benchmarks, rare ablation of individual components, the latency cost of bidirectional context, class imbalance, unseen attacks and unrealistic data splits. Based on this review, we outline a conceptual Multi-Head CNN-BiLSTM framework for secure IoV intrusion detection and propose an evaluation protocol for it.**

***Index Terms*—Internet of Vehicles, Intrusion Detection, Multi-Head CNN, BiLSTM, Deep Learning, Controller Area Network, Attention Mechanism.**

I. INTRODUCTION

Modern vehicles are networked computers. They exchange data with other vehicles, roadside infrastructure and cloud platforms, and internally they depend on buses such as the Controller Area Network (CAN) that were designed without authentication or encryption. Remote attacks on production cars have shown that this combination can be exploited [1]. In the Internet of Vehicles (IoV), attacks range from denial of service, fuzzing and message spoofing on in-vehicle networks to flooding and botnet traffic on external links. Intrusion detection systems (IDS) based on machine learning are a natural defence because they can learn the difference between normal and malicious traffic without a hand-written rule for every attack.

Deep learning has been applied to this task in several forms. A deep neural network was used to classify in-vehicle packets [2], a generative adversarial network was used to flag attacks absent from the training data [3], and a deep convolutional network was applied to CAN traffic [4]. Each family has a characteristic strength and weakness. Convolutional neural networks (CNN) are good at local patterns, such as a suspicious combination of payload bytes, but have no built-in notion of order over a long window. Recurrent networks such as the long short-term memory (LSTM) [5] model sequences, but they can miss short local structure in raw features. Hybrid CNN-LSTM models for the IoV combine the two [6], [7].

This survey is concerned with one extension of the hybrid idea. A multi-head CNN runs several convolutional branches, or "heads", in parallel, each with a different kernel width or dilation, so that short and long local patterns are extracted at the same time, an approach long used in text and image models [8], [9]. The concatenated features are passed to a bidirectional LSTM (BiLSTM) [10], which reads the sequence forwards and backwards. Attack types differ in time scale: a flooding attack shows up within a few frames, whereas a replay or spoofing attack may be visible only against a longer context, so a model that sees both scales is plausible. That plausibility is rarely tested. For a deployable IoV IDS the question is therefore not only accuracy, but also what each component contributes, how much latency the bidirectional context costs, and whether the model is small enough for in-vehicle hardware.

This paper reviews recent literature on CNN, recurrent and hybrid IDS for vehicular networks and outlines a conceptual framework, which we call the Multi-Head CNN-BiLSTM framework. Section II gives background, Section III reviews recent studies, Section IV outlines the framework, Section V compares reported performance and proposes an evaluation protocol, Section VI analyses the findings and open challenges, and Section VII concludes.

II. BACKGROUND

A. Threats and Attack Surface in the IoV

An IoV IDS faces two groups of traffic. In-vehicle traffic on the CAN bus consists of short broadcast frames, each with an identifier, a data length code and up to eight payload bytes. CAN has no sender authentication, and arbitration gives priority to the lowest identifier, so an attacker who can inject frames can flood the bus (denial of service), send random identifiers or payloads (fuzzing), or forge the content of legitimate messages (spoofing) [4], [1]. External traffic between vehicles, roadside units and cloud services is ordinary network traffic and is exposed to the flooding, botnet and intrusion attacks found in general-purpose IDS datasets [11]. Three further properties make detection harder. Attack frames are often a small share of the traffic, so the classes are imbalanced. New attacks appear that are absent from the training data. And the detector must respond within the time budget of a moving vehicle, on hardware much weaker than a server.

B. Building Blocks

Convolutional layers. A one-dimensional convolution slides a small kernel along a window of frames or features and responds to local patterns. Using several kernel widths in parallel, as in
[8] and in the Inception modules of [9], lets one layer capture patterns of different lengths. In this paper a *head* means one such parallel convolutional branch, and it should not be confused with a head of multi-head self-attention. Recurrent layers. The LSTM uses gates to keep or discard information across many steps [5]. A BiLSTM runs two LSTMs, one forward and one backward, and concatenates their states, so that each position is represented using both earlier and later frames in the window [10]. Attention. Attention weights let a model emphasise the most informative positions in a sequence instead of relying on the final state alone [12]. Training aids. Batch normalisation [13] and dropout [14] stabilise and regularise training, Adam [15] is the usual optimiser, and class imbalance is handled by re-weighting the loss, for example with focal loss [16], or by oversampling such as SMOTE [17].

C. Datasets

Four public datasets recur in the reviewed work. The Car-Hacking dataset contains CAN traffic logged from a real vehicle with injected denial-of-service, fuzzy and spoofing attacks [4]. CIC-IDS2017 contains labelled network flows covering attacks such as brute force, denial of service and botnets [11]. UNSW-NB15 is a general network intrusion dataset with several attack families [18]. The CICIoV2024 dataset, used in [19], targets vehicular attack scenarios, and IoT botnet studies also use N-BaIoT [20]. Two cautions apply. Most vehicular datasets come from a single vehicle [21]. In addition, when long recordings are cut into overlapping windows, windows from the same recording can land in both the training and the test set, which inflates reported accuracy.

III. LITERATURE REVIEW

The studies below were selected for their use of CNN, recurrent or hybrid deep learning for intrusion detection in vehicular networks, together with closely related IoT work that uses CNN-BiLSTM or attention. Their reported figures are taken from their abstracts and have not been reproduced here.

A. EARLY DEEP LEARNING IDS FOR IN-VEHICLE NETWORKS

[2] Kang and Kang, in *Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security*, train a deep neural network on in-vehicle network packets to separate normal from attack traffic. It is an early demonstration that a learned detector can replace hand-written rules on the CAN bus.

[3] Seo, Song and Kim, in *GIDS: GAN Based Intrusion Detection System for In-Vehicle Network*, use a generative adversarial network to train detectors that can also flag attacks not present in the training data. This is relevant because unseen attacks, not only known ones, are the realistic threat.

[4] Song, Woo and Kim, in *In-Vehicle Network Intrusion Detection Using Deep Convolutional Neural Network*, apply a deep CNN to CAN traffic. The Car-Hacking dataset described there is used by many later studies, so the work matters twice: it shows that convolution is effective on CAN data, and it supplies a benchmark on which later hybrid models are compared.

[22] Kukkala, Thiruloga and Pasricha, in *LATTE:*

LSTM Self-Attention Based Anomaly Detection in Embedded Automotive Platforms, use a stacked LSTM predictor with attention to learn normal CAN behaviour at design time, and a separate detection scheme to flag deviations at run time. The authors also describe changes to vehicle communication controllers that would allow the system to run on a real ECU. It is the closest reviewed design to the recurrent-plus-attention part of the framework outlined here.

B. HYBRID CNN-RECURRENT IDS FOR THE IoV

[7] Ullah et al., in *HDL-IDS: A Hybrid Deep Learning Architecture for Intrusion Detection in the Internet of Vehicles*, combine LSTM and gated recurrent unit (GRU) layers. The model was tested on a combined DDoS dataset built from CIC DoS, CIC-IDS2017 and CSE-CIC-IDS2018, and on the Car-Hacking dataset, and the authors report detection accuracy of 99.5% and 99.9% respectively. It is a clear example of a hybrid recurrent design on both external and in-vehicle traffic, although it does not use convolutional feature extraction.

[6] Alferaidi et al., in *Distributed Deep CNN-LSTM Model for Intrusion Detection Method in IoT-Based Vehicles*, combine a CNN for feature extraction with an LSTM network in a distributed implementation on Apache Spark, aimed at large-scale vehicular traffic. They report accuracy of up to 99.7%. It is the closest reviewed design to the CNN-recurrent core of the framework, but the abstract does not describe parallel convolutional branches or bidirectional recurrence.

C. CNN-BiLSTM AND ATTENTION MODELS IN RELATED IoT SETTINGS

[23] Jouhari and Guizani, in *Lightweight CNN-BiLSTM Based Intrusion Detection Systems for Resource-Constrained IoT Devices*, integrate lightweight CNNs with BiLSTM networks, evaluated on UNSW-NB15, with the aim of accurate detection at reduced model complexity. The work supports the premise that a CNN-BiLSTM combination can be kept small, although it targets general IoT traffic and not vehicles.

[20] Naeem et al., in *Efficient IoT Intrusion Detection with an Improved Attention-Based CNN-BiLSTM Architecture*, combine a 1D CNN, a BiLSTM and an attention mechanism for botnet detection, evaluated on the N-BaIoT dataset. The authors assign each part a role: the CNN captures local patterns, the BiLSTM captures long-range dependencies, and attention focuses on the important inputs. This division of labour is the one adopted in the framework below. Neither of these two studies is vehicle-specific, and testing the same design on CAN and IoV flows is the gap this survey identifies.

D. FEDERATED AND LIGHTWEIGHT DEPLOYMENT

Because vehicles hold sensitive data and have limited hardware, several works pair detection models with federated learning (FL), in which vehicles train locally and share only model updates [24]. The systematic review by Hernández-Ramos and colleagues [21] analysed 104 FL-based IDS studies. About three quarters use supervised learning, a large part rely on datasets that predate 2017, most vehicular datasets come from a single vehicle so that client splits are artificial, and most works assume the same model on every client.

[19] Zha et al., in *Lightweight Privacy-Preserving IDS for ITS: Integrated Federated Learning and Blockchain*, show that a very small model can retain high accuracy: 97.10% on CICIDS2017 with 11,904 parameters, 99.10% on Car-Hacking with 9,989 parameters and 99.65% on CICIoV2024 with 9,861 parameters, with memory use under 0.05 MB.

[25] Althunayyan, Javed and Rana, in *A Robust Multi-Stage Intrusion Detection System for In-Vehicle Network Security using Hierarchical Federated Learning*, focus on the CAN bus. Their multi-stage IDS detects known attacks and also defends against unseen ones, with reported F1-scores above 0.99 for seen attacks, above 0.95 for novel attacks and a detection rate of 99.99%. Both studies are useful as reference points for model size and for unseen-attack performance.

IV. THE MULTI-HEAD CNN-BiLSTM FRAMEWORK

Building on the reviewed work, we outline a conceptual Multi-Head CNN-BiLSTM framework for secure intrusion detection in the IoV. It is a design proposal derived from this survey. It has not been implemented or evaluated here, and no performance claims are made for it. "Secure" is used in two senses: the model protects the vehicle network, and its design takes account of the deployment limits and weaknesses of the IDS itself.

1. Input and preprocessing. CAN traffic is cut into windows of N consecutive frames, each described by its identifier, data length, payload bytes and the time since the previous frame. Network flows are described by their flow statistics. Features are normalised with training-set statistics only, and each window is labelled as normal or with its attack type.

Class imbalance is handled by loss weighting or focal loss [16], with oversampling such as SMOTE [17] applied to the training split only.

2. Multi-head CNN. Three or four parallel 1D convolutional heads process each window. Small kernels (for example width 3) capture byte-level and inter-frame patterns, and larger or dilated kernels capture longer patterns [8], [9]. Each head applies convolution, batch normalisation [13] and a non-linearity, with "same" padding so that the head outputs align in time and can be concatenated along the feature axis. Heads may also be assigned to separate feature groups, such as payload bytes and timing.
3. BiLSTM encoder. The concatenated feature sequence is passed to a BiLSTM [10] whose forward and backward states are joined at each position. A single layer with a modest hidden size is the starting point, because the aim is a small model [23], [19].
4. Attention pooling and classifier. An attention layer [12] weights the BiLSTM outputs so that the frames that matter most dominate the summary vector, and the weights give a rough, not guaranteed, indication of which frames drove an alert. A dense layer with dropout [14] and a softmax output classifies the window as normal or as one of the attack classes.
5. Training. The model is trained with Adam [15] and early stopping on a validation set. Head count, kernel widths, hidden size and window length are chosen on the validation split and reported in full.
6. Deployment. The trained model is compressed by pruning, quantisation or knowledge distillation, as in [19], and run on an ECU gateway or roadside unit. Because the BiLSTM needs the whole window, detection delay includes the time to collect it, so window length is a security parameter and must be reported. Where raw data cannot leave the vehicle, the same model can be trained with federated learning [24], [21].
7. Evaluation protocol. We recommend testing on vehicle-specific and network datasets, with splits that keep windows from one recording together and with held-out attack classes, and reporting detection quality, efficiency and the ablations listed in Table II.

V. PERFORMANCE METRICS

Table I compares the reviewed studies. Values are those reported by the authors and are not directly comparable, because datasets, attack types, window definitions and validation protocols differ. Entries marked "Not reported" were not stated in the sources consulted. Table II lists the metrics and ablations we propose for evaluating the framework in Section IV.

TABLE I. COMPARISON OF REVIEWED DEEP LEARNING IDS STUDIES FOR VEHICULAR AND RELATED NETWORKS

Study	Setting	Datasets	Model / mechanism	Reported result
Song et al. [4]	In-vehicle CAN bus	Car-Hacking	Deep CNN on CAN traffic	Not reported
Kukkala et al. [22] (LATTE)	In-vehicle CAN, embedded ECU	CAN attack scenarios (not itemised)	Stacked LSTM predictor with attention; run-time anomaly detection	Not reported
Ullah et al. [7] (HDL-IDS)	IoV, external and in-vehicle	Combined DDoS set (CIC DoS, CIC-IDS2017, CSE-CIC-IDS2018); Car-Hacking	Hybrid LSTM + GRU	99.5% (DDoS) and 99.9% (Car-Hacking) accuracy
Alferaidi et al. [6]	IoV, distributed on Apache Spark	Not itemised in the abstract	CNN + LSTM	Accuracy up to 99.7%
Jouhari and Guizani [23]	Resource-constrained IoT	UNSW-NB15	Lightweight CNN + BiLSTM	Not reported here
Naeem et al. [20]	IoT botnet detection	N-BaIoT	1D CNN + BiLSTM + attention	Not reported here
Zha et al. [19]	ITS / IoV, resource-limited vehicles	CICIDS2017; Car-Hacking; CICIoV2024	Lightweight model with knowledge distillation (with FL and blockchain)	97.10%, 99.10% and 99.65% accuracy; about 10,000 parameters

Althunayyan et al. [25]	In-vehicle CAN, hierarchical FL	CAN-bus data	Multi-stage IDS for seen and unseen attacks	F1 above 0.99 (seen), above 0.95 (novel); detection rate 99.99%

TABLE II. PROPOSED EVALUATION PROTOCOL FOR THE MULTI-HEAD CNN-BiLSTM FRAMEWORK

Aspect	What to measure	Variants / setting	Reason
Detection quality	Per-class precision, recall and F1; macro-F1; false positive rate	Report each attack class, not only overall accuracy	Accuracy alone hides failures on rare attack classes
Efficiency	Parameters, model size, inference time per window, detection delay including window collection	Measured on an embedded CPU or automotive-grade board	Shows whether the model fits in-vehicle limits
Multi-head ablation	Single-kernel CNN against 2, 3 and 4 heads	Same total parameter budget where possible	Tests whether parallel scales help or only add parameters
Recurrent ablation	CNN only; LSTM; BiLSTM; GRU	Same CNN front end	Isolates the contribution of backward context
Attention ablation	With and without attention pooling	Same backbone	Tests whether attention adds accuracy or only interpretability
Data protocol	Random split; recording-level split; cross-vehicle split	Windows of one recording never in both training and test sets	Avoids leakage from overlapping windows [21]
Robustness	Held-out attack class; noisy or missing frames; shorter windows	Train without the class, test with it	Estimates generalisation to unseen attacks [3], [25]

VI. ANALYSIS

Several patterns emerge from the reviewed work.

Accuracy is near ceiling, so the differentiators lie elsewhere. Reported accuracies on CAN and IoV benchmarks reach 99% or more [7], [6], [19]. When results are this close to ceiling, further gains are small and may fall within the variation caused by the data split. What separates studies is model size, latency, behaviour on rare and unseen attacks and the validity of the evaluation. The efficiency result in [19], high accuracy with roughly ten thousand parameters, is a good example.

Hybrids are common, but their components are rarely isolated. CNN-LSTM, LSTM-GRU and CNN-BiLSTM designs [7], [6], [23], [20] are usually compared with other models, not with versions of themselves with one component removed. The abstracts we consulted do not state how much the convolution, the bidirectional recurrence or the attention each contributes. The benefit of parallel convolutional heads in particular is supported by evidence from text and image models [8], [9] and not by vehicular IDS studies. The ablations in Table II are meant to close this gap.

Bidirectional context has a latency cost. A BiLSTM needs the complete window before it can classify it, because its backward pass starts from the last frame. Detection delay is therefore at least the window duration, and an attack shorter than one window may be noticed late. The abstracts consulted do not report detection delay, even though it matters for a moving vehicle.

Imbalance and unseen attacks remain hard. Attack traffic is often a minority, and a detector trained on a fixed list of attacks may miss new ones. Loss re-weighting [16] and oversampling

[17] address imbalance. Unseen attacks are targeted by generative [3], predictive [22] and multi-stage [25] designs, but it is not clear from the abstracts of the supervised hybrid classifiers that they were tested on held-out attack types.

Experimental realism is limited. The review in [21] found that many FL-based IDS studies use outdated datasets and artificial client splits, and that most vehicular datasets come from one vehicle. Together with overlapping windows, this means that random splits can overstate real-world performance.

Open challenges. (i) Realistic multi-vehicle CAN datasets with public benchmark splits. (ii) Ablation studies that isolate the multi-head, bidirectional and attention components. (iii) Reporting of detection delay and model size alongside accuracy. (iv) Detection of unseen attacks. (v) Robustness of the IDS itself against inputs crafted to evade it. (vi) Explainable alerts that a security analyst can act on.

Future directions. Promising directions include testing the framework on CAN and network data together, causal or streaming recurrent variants that do not wait for a full window, transformer or temporal-convolution alternatives to the recurrent layer [12], semi-supervised and unsupervised

detectors for unknown attacks, federated training across vehicles [24], explanation of alerts with attribution methods [26], and field tests on real vehicles.

VII. CONCLUSION

Deep learning gives the Internet of Vehicles a practical way to detect attacks on both in-vehicle and external networks. The reviewed studies show that CNNs work on CAN data, that recurrent and attention models capture its temporal structure, that hybrid designs reach accuracy of about 99% on common benchmarks, and that very small models can keep that accuracy. At the same time, the evidence is limited by near-saturated benchmarks, a lack of ablation studies, unreported detection delay, unrealistic data splits and little testing on unseen attacks. A framework such as the proposed Multi-Head CNN-BiLSTM design, which extracts features at several scales, reads them in both directions and pools them with attention, is a reasonable candidate for addressing these gaps in principle, and needs to be implemented and tested under the evaluation protocol outlined here.

REFERENCES

[1] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, 2015.

[2] 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, Art. e0155781, 2016.

[3] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in Proc. 16th Annu. Conf. Privacy, Security and Trust (PST), 2018, pp. 1–6.

[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. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] A. Alferaidi, K. Yadav, Y. Alharbi, N. Razmjooy, W. Viriyasitavat, K. Gulati, S. Kautish, G. Dhiman, and R. Ranjbarzadeh, "Distributed deep CNN-LSTM model for intrusion detection method in IoT-based vehicles," Mathematical Problems in Engineering, vol. 2022, pp. 1–8, 2022. doi: 10.1155/2022/3424819.

[7] S. Ullah, M. A. Khan, J. Ahmad, S. S. Jamal, Z. e Huma, M. T. Hassan, N. Pitropakis, Arshad, and W. J. Buchanan, "HDL-IDS: A hybrid deep learning architecture for intrusion detection in the Internet of Vehicles," Sensors, vol. 22, no. 4, Art. 1340, 2022. doi: 10.3390/s22041340.

[8] Y. Kim, "Convolutional neural networks for sentence classification," in Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.

[9] C. Szegedy et al., "Going deeper with convolutions," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2015,pp. 1–9.

[10] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Trans. Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

[11] 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. Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116. doi: 10.5220/0006639801080116.

[12] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems 30 (NeurIPS), 2017, pp. 5998–6008.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proc. 32nd Int. Conf. Machine Learning (ICML), PMLR vol. 37, 2015,pp. 448–456.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 2980–2988.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[18] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. Military Communications and Information Systems Conf. (MilCIS), 2015,pp. 1–6.

[19] J. Zha, G. Zhang, L. Jin, S. Dang, and W. Duan, "Lightweight privacy-preserving IDS for ITS:

Integrated federated learning and blockchain," IEEE Trans. Intell. Transp. Syst., vol. 27, no. 7,pp. 8739–8754, 2026. doi: 10.1109/TITS.2026.3661633.

[20] A. Naeem et al., "Efficient IoT intrusion detection with an improved attention-based CNN-BiLSTM architecture," arXiv:2503.19339, 2025.

[21] J. L. Hernández-Ramos, G. Karopoulos, E. Chatzoglou, V. Kouliaridis, E. Marmol, A. Gonzalez-Vidal, and G. Kambourakis, "Intrusion detection based on federated learning: A systematic review," arXiv:2308.09522, 2023.

[22] V. K. Kukkala, S. V. Thiruloga, and S. Pasricha, "LATTE: LSTM self-attention based anomaly detection in embedded automotive platforms," arXiv:2107.05561, 2021.

[23] M. Jouhari and M. Guizani, "Lightweight CNN-BiLSTM based intrusion detection systems for resource-constrained IoT devices," arXiv:2406.02768, 2024.

[24] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR vol. 54, 2017, pp. 1273–1282.

[25] M. Althunayyan, A. Javed, and O. Rana, "A robust multi-stage intrusion detection system for in-vehicle network security using hierarchical federated learning," arXiv:2408.08433, 2024.

[26] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems 30 (NeurIPS), 2017, pp. 4765–4774.