Network Communication and Electronic Control Strategy of New Energy Vehicles Based on Cloud Platform in the IoT Environment

Hong Xiao, School of Intelligent Manufacturing and Transportation, Urban Vocational College of Sichuan, China

Yufeng Tang, School of Mechanical Engineering, Sichuan University of Science and Engineering, China*

ABSTRACT

Aiming at the low accuracy of network intrusion detection (In-De) in the traditional network communication strategy of new energy vehicles (NEVs), this paper proposes an electronic control (E-C) strategy for network communication of NEVs based on cloud platform in the internet of things (IoT) environment. First, based on the cloud platform and deep learning (D-L) algorithm, the E-C system model including sensor, actuator, gateway, and cloud platform is constructed, and on this basis, the edge computing model is introduced to efficiently handle information interaction and computing tasks. Then, by using Bi-LSTM neural network to train historical data in the cloud center layer of the system, a D-L method combining cloud and edge nodes is proposed. Finally, by introducing the AlexNet network into the model, the problem of gradient vanishing when the network is deep is solved and the training speed is accelerated.

KEYWORDS

AlexNet Network, Cloud Platform, Internet of Things, Network Communications, NEVs

INTRODUCTION

In today's society, the development of NEVs is particularly important. Though NEVs use new energy power devices and other new technologies, the technology is not as mature as traditional vehicles (Habibi et al., 2019; Madria et al., 2020; Qiu et al., 2021). To reduce the possible risks and potential threats of NEVs, some scholars (Habibi et al., 2019) have made changes to the electronic control strategy of traditional NEVs network communication to increase the intervention for the network communication of new energy vehicles and improve the intelligence of communication. Guided by the functional requirements of NEVs in the IoT environment, new functions have been added on the basis of the cloud platform (Abayomi et al., 2020; Culler, 2017; He, 2021) such as wireless communication and vehicle fault analysis. These new functions increase the performance of the IoT platform.

DOI: 10.4018/ijaci.318135

*Corresponding Author

Wide application of NEVs becoming inseparable from the internet, however, leads to greater risks. The scale of NEV users connected to the internet is becoming larger and larger, and the massive network resource data has brought about extremely serious network security problems, especially that of IoT communication (Riyaz & Ganapathy, 2020; Yin et al., 2020; Zavrak & Iskefiyeli, 2020). It is important to enhance the ability of In-De system to detect network behaviors and enable the system to better achieve autonomous defense (Lu et al., 2019; Zhang et al., 2018). In-De system is an active defensive means to significantly enhance network security. It collects and analyzes information from computers and various regions of the network, identifies normal behaviors and abnormal behaviors that threaten network security in real time, and responds to abnormal data detected in real time (Sepas-moghaddam et al., 2021; Tamy et al., 2019; Yao et al., 2021). In addition, IoT equipment has limited power, and data transmission will cause large power consumption of terminal equipment. How to use the bandwidth and computing resources of existing cloud computing to efficiently access these data is presently one of the hot topics in the industry (Ashiku & Dagli, 2021; Chen et al., 2021; Zhang et al., 2021).

It is urgent to improve the accuracy and efficiency of communication in NEVs to provide direction for more research in order to solve the current issues in the IoT environment and greatly improve reliability.

RELATED RESEARCH

Throughout previous studies, traditional machine learning algorithms have relied too much on feature engineering and selection. With the continuous increase of network data, more and more scholars and researchers are integrating new IoT frameworks to balance service and data processing requirements by accessing edge computing nodes or devices for the IoT.

Oma et al. (2018) added edge nodes as the middle layer, through pushing sensor data directly to processing and used edge nodes to process feedback data to reduce the delay. Edge nodes, however, are limited by computing capacity.

Datta et al. (2019) proposed an IoT edge computing architecture that combined the relay computing layer to process IoT. This method used virtual IoT devices to process local data and improved the real-time response speed of data. The method, however, didn't account for the increase in data demand and that the surge in sensing equipment would be limited by communication and power and computing capabilities.

Li et al. (2020) modeled the computing offload process as the minimum allocatable wireless resource block level and proposed a method for computing offload. This method measured the cost-effectiveness of resource allocation and energy conservation but could not meet the requirements of real-time and accuracy.

Bhattacharya & Lane (2020) proposed a smart wearable device model by integrating the key role of smart devices in IoT systems. The purpose was to promote accuracy of models in audio recognition tasks, but this method has poor anti-interference performance.

Aimed at the task allocation problem of the IoT system, Alsheikh et al. (2021) proposed an architecture for intelligent mobile devices and cloud centers to process sensing data and computing tasks in collaboration. The purpose of this method was to effectively reduce cloud pressure and improve system efficiency, but it turned out to be inefficient.

Lu et al. (2020) proposed a network In-De method combined with transfer learning by using an unsupervised deep auto-encoder. The method uses the Softmax regression model to encode and classify the label information in the source domain, which greatly improves the accuracy of information, but this method cannot avoid the disadvantage of single point deployment, and the server resource utilization is low.

According to the characteristics of network traffic, based on federated learning, a new network intrusion detection method was proposed by Tang et al., (2022) by analyzing the distribution characteristics of network data. This method improved the detection accuracy and protected privacy in network traffic; although, it takes up a large amount of storage space.

Aimed at the low precision of network In-De in network communication strategy of NEVs, this study proposes an electronic control strategy for network communication of NEVs with the cloud platform. Basic ideas are as follows: 1) build an electronic control system model based on the cloud platform and a D-L algorithm and introduce an edge computing model, 2) the cloud and edge nodes are combined through a D-L method, and 3) the AlexNet network is introduced into the model for optimization.

The innovations of the proposed method lie in: 1) the edge computing model is introduced to deal with the information interaction and computing tasks in the process of model computing and 2) the proposed D-L method with cloud and edge nodes can train historical data through Bi-LSTM neural network in the cloud center layer of the system.

ELECTRONIC CONTROL STRATEGY MODEL BASED ON THE D-L CLOUD PLATFORM

Overall Structure of Electronic Control System

Figure 1 shows the overall framework of the D-L electronic control system model with a cloud platform. The main function of its communication system is to realize remote monitoring and optimal calibration of NEV. The electronic control unit of NEV is directly used as a vehicle data acquisition device, and its wireless communication function is used to realize data communication with mobile phones and edge servers, forming a dual monitoring mode of *mobile phone + cloud* for vehicles. The edge server can analyze and process a large amount of vehicle data uploaded by the electronic control unit, generate optimal control parameters, and remotely optimize and calibrate the electronic control unit. The incoming NEV data goes through the data center and then combines with the characteristics of the network model in the cloud and edge layer. The data then embeds in the edge layer deep learning model for deep training and improves the accuracy and efficiency of communication.

Overall Model Architecture

The entire system consists of sensors, actuators, gateways, and a cloud platform, as shown in Figure 2. Sensors and actuators transmit data wirelessly through gateways, and communication methods mainly include 2.4 GHz and 4.33 MHz. The data is combined with the cloud-edge collaboration to process the communication data, and the deep learning method is embedded in the cloud-edge layer to improve the accuracy and efficiency of the data.

The intelligent energy-saving control gateway is equipped with an infrared transmitter and receiver for remote control of the control module and sensor module of the NEVs. It uses wireless communication to obtain the measurement data of the control module and sensor module. It uses these measurement data for rule calculation and the infrared transmitter and power controller to control the control module and sensor module.

The cloud platform is composed of two parts: IoT components and data processing servers. The IoT component is responsible for data and message routing; data processing servers provide various data analysis and modeling operations including data presentation and background management.

Edge Computing Model

Edge computing technology plays a vital role in today's society; it (Tang et al., 2022) has applications in many fields as well as smart city networks. For many application tasks with low latency requirements, edge computing technology can provide a wider range of processing solutions, so that information interaction and computing tasks can be processed in a more timely and efficient manner. For the application server to accurately identify the location, the method of combining edge computing technology to process the task has greater advantages than that of only using the cloud center to process the task. The location information can be directly responded to and then calculated through the edge node. The edge computing model is combined with cloud data interaction and processing, shown in Figure 3.

Figure 1. D-L electronic control system based on the cloud platform

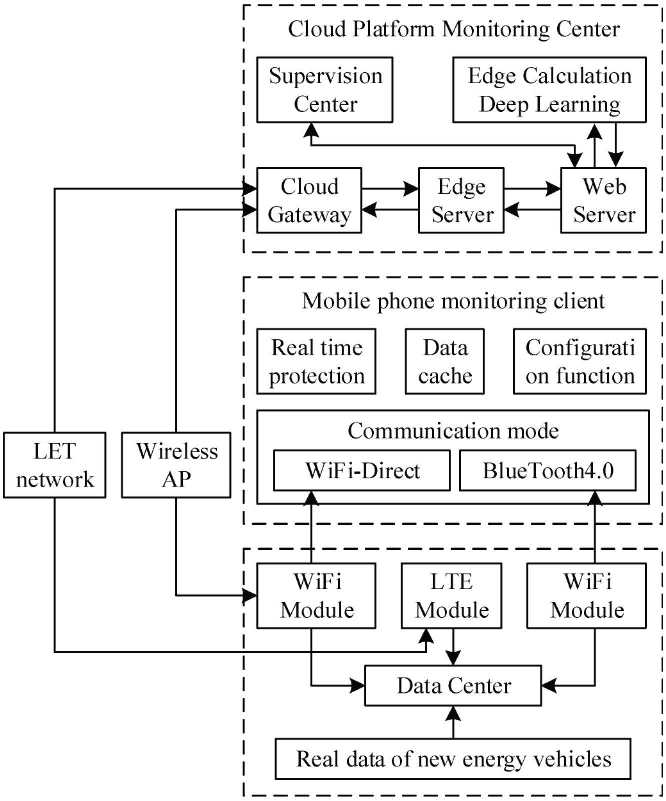


Figure 2. Structure of the electronic control model of the network communication of NEVs

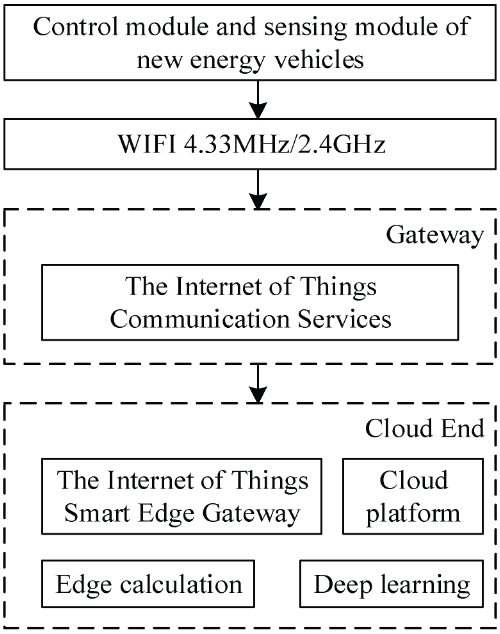
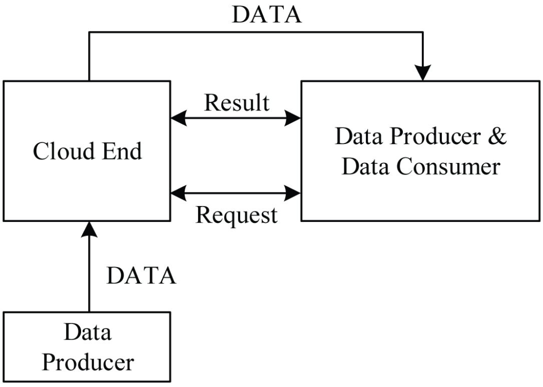

Control module and sensing module of new energy vehicles
WIFI 4.33MHz/2.4GHz
Gateway
The Internet of Things Communication Services
Cloud End
The Internet of Things Smart Edge Gateway
Cloud platform
Edge calculation
Deep learning

Figure 3. Edge calculation model

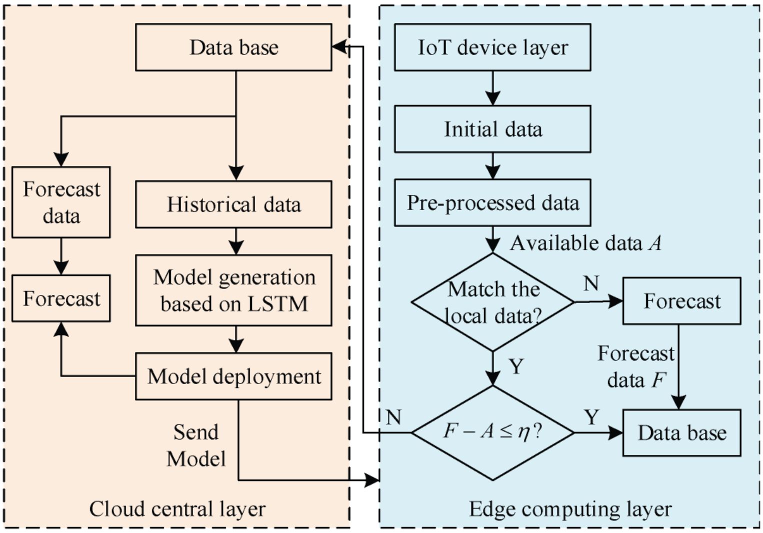


In the edge computing processing model, the intelligent device close to the data source acts as the data producer and sends the tasks and data parts after normalizing the data. In this model, the edge node and the cloud center reasonably allocate tasks and data storage and conduct two-way information response and data interaction.

Edge-Cloud Interaction Architecture

Proposed here is a D-L method with cloud and edge nodes. In the cloud center layer of the system, Bi-LSTM is used to train the former. The resulting model is deployed as input data. The edge computing layer receives the original data $A\left(a_1, a_2, \ldots, a_n\right)$ from the device layer at any time t_0, and then performs preprocessing and detects whether there is matching data in the local database for a period of time. The edge layer will package and publish the real data to the cloud center layer according to the application needs, and then use the new data to train and update the model. The process of edge-cloud collaboration method is shown in Figure 4.

Figure 4. The process of the edge-cloud collaboration method



AlexNet D-L Model

The AlexNet network can combine commonly used processing methods of the current network such as Rectified Linear Units (ReLU) function as the activation function, graphics processing unit (GPU) for parallel processing of data, dropout technology to prevent over-fitting, and local response normalization (LRN) function and image enhancement technology.

Older computer hardware cannot meet the computing power requirements of this network model, so the network model designer divides the network into two parts, and the training process is carried out on two GPUs. Since the two GPUs will have data isolation phenomena when they perform operations separately, it is necessary to fuse the data of the two GPUs at some layers. The framework of AlexNet is shown in Figure 5.

Long-Term Evolution (LTE) Architecture

The Long-Term Evolution (LTE) architecture is shown in Figure 6. The Evolved Universal Terrestrial Radio Access Network (EUTRAN) is composed of Evolved Node B (eNB), and the connection between EUTRAN and the core network has been simplified. Ports S1 and S2 are connected to the eNB via an Evolved Packet Core (EPC) network. The EPC and EUTRAN are connected via port S1, while eNB and eNB are connected via port X2. The structure of EUTRAN itself also adopts a layered model, which is divided into a transmission network layer and a wireless network layer.

Figure 5. Architecture of the AlexNet network

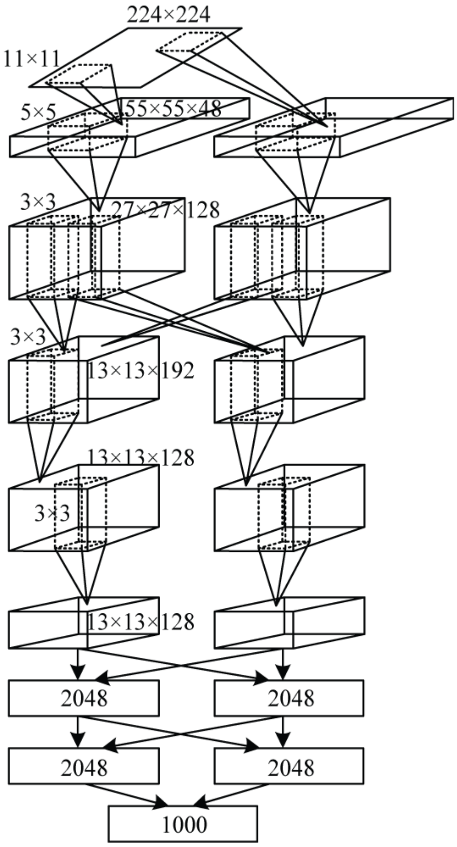
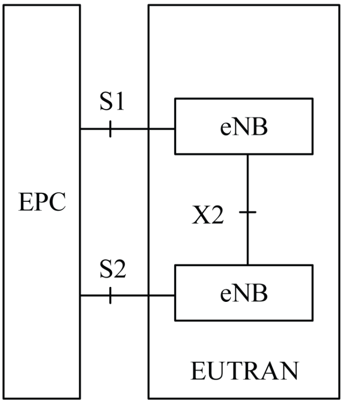


Figure 6. Long-term evolution architecture



EXPERIMENT AND ANALYSIS

Experimental Environment

The experiments use the Python D-L framework, Python 3.5.7 programming language, and the edge-cloud collaborative reasoning model based on the current widely used deep neural network VGG16. The data set used in experiments is the CIFAR10, which is used in image classification. Raspberry Pi 4B is used as the edge device. It can represent the typical computing power of most edge devices at present. The NVIDIA GPU with GEFORCE RTX 2080Ti is used as the cloud server. See Table 1 and Table 2 for the detailed configurations of the two devices.

Datasets

Three data sets, NSL-KDD, UNSW-NB15, and CICIDS-2017, are used for the experiment.

The NSL-KDD dataset includes a 125 973 records training set and a 22 543 records test set. This dataset contains four types of attacks, of which the amount of data is much lower than the amount of normal type data. The details are shown in Table 3.

Table 1. Edge device platform configuration

Name	Configuration
Model	Raspberry Pi 4B
CPU	4×ARM Cortex-A72 @ 1.5GHz
RAM	4GB LPDDR4

Table 2. Cloud server platform configuration

Name	Configuration
Model	GEFORCE RTX 2080Ti
CPU	Intel i7-9700K 4.20GHz
GPU	NGR 2080Ti
RAM	11GB GDDR6 14000MHZ

Table 3. NSL-KDD dataset information

Type	Training set	Test set
Normal	73282	10538
Dos	53859	8579
Probe	14752	3759
R2L	1053	3472
U2R	96	330

The UNSW-NB15 is from the Network Range Laboratory of the Australian Network Security Center. The UNSW-NB15 dataset has a 175 211 records training set and a 82 286 records test set. The details are shown in Table 4.

The CICDS-2017 dataset contains network traffic based on packet and the two-way flow format, and each record contains 82 network flow characteristics. It includes a wider range of attack types. The details are shown in Table 5.

Evaluation Indices

To more accurately evaluate the quality of the model, five evaluation indices are selected: Accuracy A, Detection rate (D), Recall rate (R), F-1 value, and False Positive Rate (FPR).

Table 4. UNSW-NB15 information

Type	Training set	Test set
Normal	58643	38547
Generic	49538	21095
Exploits	32905	13827
Fuzzers	20194	6692
Dos	13286	4219
Reconnaissance	111365	4029
Analysis	2037	772
Backdoor	1988	603
Shellcode	1312	486

Table 5. CICIDS-2017 dataset information

Type	Training set	Test set
Normal	1539825	1029478
Dos	174297	127439
PortScan	105382	74092
DDos	80396	59382
Pataor	10458	6309
Web attack	1732	1132
Botnet	1549	975
Infiltration	35	22
Heartbleed	10	4

Accuracy A is the percentage of data correctly classified, which is calculated as formula (1):

$$A = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{1}$$

Detection rate D is the proportion of correctly classified normal data among the actual normal data, which is calculated as formula (2):

$$D = \frac{T_P}{T_P + F_P} \tag{2}$$

Recall rate R is the proportion of correctly classified normal data among all classified normal data, which is calculated as formula (3):

$$R = \frac{T_P}{T_P + F_N} \tag{3}$$

F-1 can be calculated as shown in formula (4):

$$F1 = \frac{2DR}{D + R} \tag{4}$$

The FPR is the proportion of correctly classified normal data among all classified normal data, which is calculated as formula (5):

$$FPR = \frac{T_P}{F_N + T_P} \tag{5}$$

where T_P is the quantity of data that the classifier classifies the attack data into attack types, T_N is the quantity of data that the classifier classifies normal data into normal types. F_P is the quantity of data that the classifier classifies normal data into attack types. F_N is the quantity of data that the classifier classifies attack data into normal types.

Relationship Between Model Training Times and Accuracy

AlexNet is trained to obtain the prediction classification of each sample, and the accuracy is obtained by calculating the proportion of correct quantity of prediction classification to all samples. The simulation function is called to output the list of accuracy and running time. The accuracy of AlexNet after training is shown in Figure 7.

In the deployment phase, the trained models are deployed to the Raspberry Pie and the computer respectively. The time required under the same classification accuracy is compared. The accuracy and running time of AlexNet network classification are shown in Figure 8.

From Figures 7 and 8, we can see that when the epoch reaches 10, the training accuracy stabilizes and fluctuates above 95%. As time goes on, the training accuracy of the model keeps rising. When

Figure 7. AlexNet network training accuracy

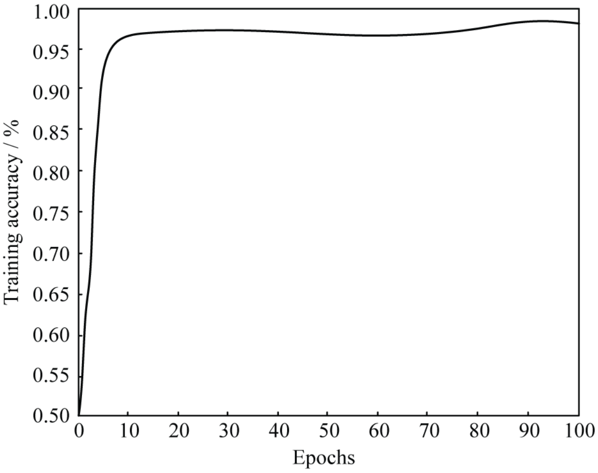
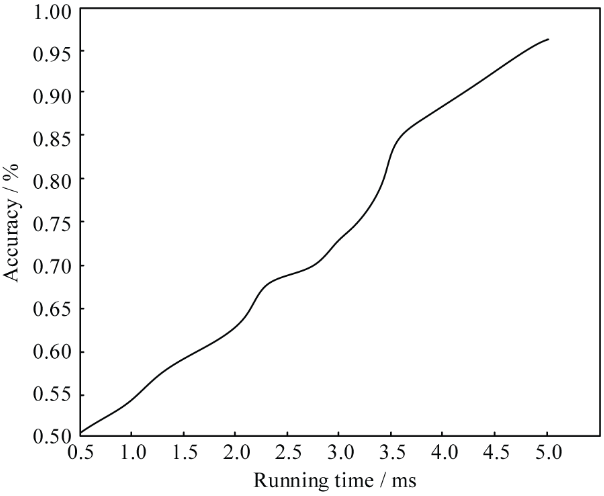


Figure 8. Accuracy and running time of AlexNet network classification



the time reaches 5ms, the accuracy has reached over 95%, which shows that the model converges faster and has better accuracy.

Loss Comparison of Different Batch Sizes

In essence, the optimization of the neural network is the compromise between the average performance and stability of the model, that is, the deviation between the models obtained by repeated training. The most ideal result is to obtain a model with small average error and strong stability, which means that the model is good and easy to repeat. Figures 9, 10 and 11 show the training loss under different batch sizes.

From the above three Figures, we can see that when the value of batch size changes, the loss is not obvious. In the case of 32, the difference between training loss and test loss is large, and it is difficult to achieve an accurate prediction. When batch size is 128, the test loss is always lower than the training. Therefore, it is optimal to set its value to 64.

Figure 9. Loss of the model when batch size is 32

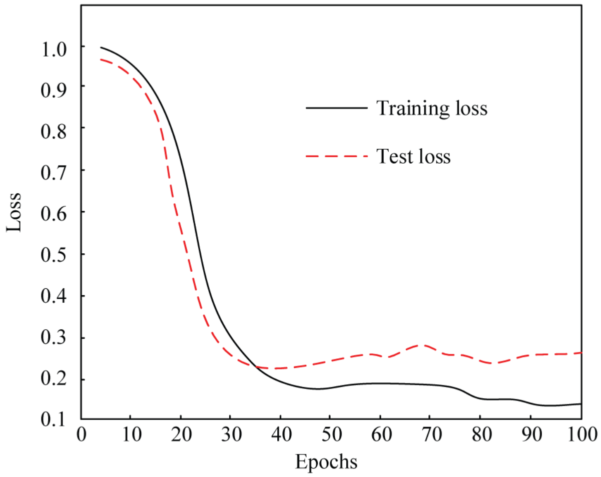
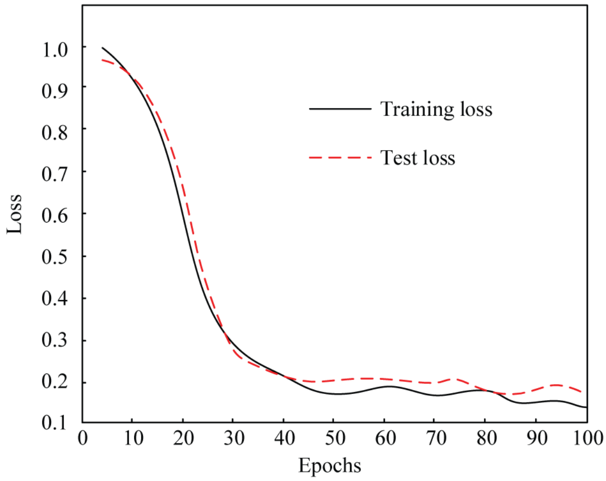


Figure 10. Loss of the model when batch size is 64



Comparative Analysis

For the purpose of verifying the superiority of the network communication electronic control strategy for NEVs, the proposed method and reference 19], reference 23], and reference 24] are compared and analyzed under NSL-KDD dataset, UNSW-NB15 dataset, and CICIDS-2017 dataset. Final calculation values of each are shown in Tables 6, 7 and 8.

It can be seen from Table 6, Table 7, and Table 8, that under the condition of using three different datasets, the proposed network communication electronic control strategy for NEVs is superior to the other three comparison methods in five evaluation indices. The lowest accuracy of the proposed algorithm is 94.45%, the highest accuracy is 95.37%; the lowest F1 value is 91.45%, and the highest F1 value is 92.35%. Compared with the other three comparison algorithms, the accuracy and comprehensive performance F1 value of the proposed algorithm have increased. This is because the Bi-LSTM network can eliminate the problem of gradient vanishing and reduce the time interval

Figure 11. Loss of the model when batch size is 128

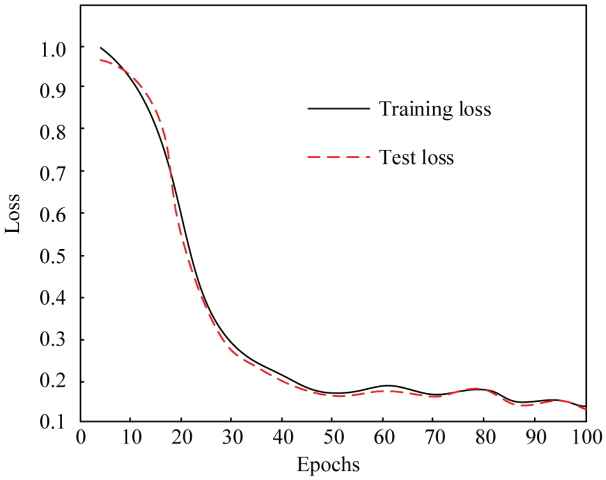


Table 6. Calculation results when using NSL-KDD dataset

Index	Method			
	Proposed method	Ref. 19	Ref. 23	Ref. 24
A	95.34%	89.67%	86.84%	82.49%
D	92.67%	87.42%	84.54%	81.31%
R	91.89%	87.33%	83.94%	81.02%
F1	92.28%	87.37%	84.24%	81.16%
FPR	94.75%	89.21%	86.49%	82.93%

Table 7. Calculation results when using UNSW-NB15 dataset

Index	Method			
	Proposed method	Ref. 19	Ref. 23	Ref. 24
A	94.45%	89.13%	85.94%	81.72%
D	92.03%	86.33%	83.89%	80.43%
R	91.02%	86.53%	83.21%	80.13%
F1	91.45%	86.43%	83.53%	80.27%
FPR	93.55%	88.71%	85.42%	82.24%

Table 8. Calculation results when using CICIDS-2017 dataset

Index	Method			
	Proposed method	Ref. 19	Ref. 23	Ref. 24
A	95.37%	89.21%	86.53%	82.64%
D	92.32%	87.04%	84.21%	81.66%
R	91.32%	87.56%	83.23%	81.44%
F1	92.35%	87.53%	84.55%	81.23%
FPR	94.06%	89.52%	86.76%	82.26%

from obtaining input to making decisions. In addition, the introduction of a double-layer attention mechanism can calculate the weight distribution of different bytes.

CONCLUSION

Aimed at the low accuracy of network In-De in the traditional network communication strategy of NEVs, this paper proposes an electronic control strategy for network communication of NEVs based on a cloud platform in the IoT environment. The experimental results show that the edge computing model can effectively improve the efficiency of information exchange in the process of model computing. Combining cloud and edge nodes based on D-L can effectively predict future data at the edge computing layer. The AlexNet network can effectively solve the problem of gradient vanishing when the network is deep and can speed up the training to a certain extent. The future work will focus on how to add a distributed learning model to the sensor device to simulate the edge data flow, instead of transmitting all the original sensor values to the edge node, to further reduce energy consumption.

AUTHOR NOTE

Hong Xiao: https://orcid.org/0000-0002-8647-1713
Yufeng Tang: https://orcid.org/0000-0003-3722-7869

The author declares that there is no competing interest for this work, and no funding was received.

Correspondence concerning this article should be addressed to Yufeng Tang, School of Mechanical Engineering, Sichuan University of Science and Engineering, Yibin, Sichuan, 644000, China. Email: 10659@suse.edu.cn.

ACKNOWLEDGMENT

The authors would like to thank the editor and anonymous reviewers for their contributions toward improving the quality of this paper. The data used to support the findings of this study are included within the article. This work supported by this research was funded by Major Hazard Measurement and Control Key Laboratory of Sichuan Province, grant number KFKT-2021-01 and the open fund of Sichuan Provincial Key Lab of Process Equipment and Control, grant number GK202207.

REFERENCES

Abayomi, A., Adebola, O. A., & Joseph, B. A. (2020). Network In-De based on D-L model optimized with rule based hybrid feature selection. *A Global Perspective, 29*(6), 223-239.

Alsheikh, M. A., Niyato, D., Lin, S., Tan, H., & Han, Z. (2021). Mobile big data analytics using D-L and apache spark. *IEEE Network*, *30*(3), 22–29. doi:10.1109/MNET.2016.7474340

Ashiku, L., & Dagli, C. (2021). Network In-De system using D-L. *Procedia Computer Science*, *185*(34), 201–224.

Bhattacharya, S., & Lane, N. D. (2020). Sparsification and separation of D-L layers for constrained resource inference on wearables. *Proceedings of ACM Conference on Embedded Network Sensor System*, 176-189.

Chen, T. T., Chen, Z. M., & Zhou, Z. X. (2021). Computational research and implementation of prediction of pork price based on D-L. *Journal of Physics: Conference Series*, *1815*(1), 32–49. doi:10.1088/1742-6596/1815/1/012032

Culler, D. E. (2017). The once and future internet of everything. *GetMobile: Mobile Computing and Communications*, *20*(3), 5–11. doi:10.1145/3036699.3036701

Datta, S. K., Bonnet, C., & Nikaein, N. (2014). An IoT gateway centric architecture to provide novel M2M services. *Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT)*, 514-519. doi:10.1109/WF-IoT.2014.6803221

Habibi, M. A., Nasimi, M., Han, B., & Schotten, H. D. (2019). A comprehensive survey of RAN architectures toward 5G mobile communication systems. *IEEE Access: Practical Innovations, Open Solutions*, *7*(5), 70371–70421. doi:10.1109/ACCESS.2019.2919657

He, X. Z. (2021). Analysis of network In-De technology based on computer information security technology. *Journal of Physics: Conference Series*, *1744*(4), 1742–1744.

Li, X., Dang, Y., Aazam, M., Peng, X., Chen, T., & Chen, C. (2020). Energy-efficient computation offloading in vehicular edge cloud computing. *Access*, *8*(12), 37632–37644. doi:10.1109/ACCESS.2020.2975310

Lu, L., Yi, Y., Huang, F., Wang, K., & Wang, Q. (2019). Integrating local CNN and global CNN for script identification in natural scene images. *IEEE Access: Practical Innovations, Open Solutions*, *7*(06), 52669–52679. doi:10.1109/ACCESS.2019.2911964

Lu, M., Du, G., & Ji, Z. (2020). Network In-De based on deep transfer learning. *Jisuanji Yingyong Yanjiu*, *37*(9), 2811–2814.

Madria, S., Kumar, V., & Dalvi, R. (2020). Sensor cloud: A cloud of virtual sensors. *IEEE Software*, *31*(2), 70–77. doi:10.1109/MS.2013.141

Oma, R., Nakamura, S., Enokido, T., & Takizawa, M. (2018). An energy-efficient model of fog and device nodes in IoT. *Proceedings of 32nd International Conference on Advanced Information Networking and Applications (AINA)*, 301-306. doi:10.1109/WAINA.2018.00102

Qiu, B., Chen, K., & He, K. (2021). Research on vehicle network In-De technology based on dynamic data set. *Proceedings of 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, 12-14.

Riyaz, B., & Ganapathy, S. (2020). A D-L approach for effective In-De in wireless networks using CNN. *Soft Computing*, *24*(22), 17265–17278. doi:10.1007/s00500-020-05017-0

Sepas-moghaddam, A., Etemad, A., Pereira, F., & Correia, P. L. (2021). Long short-term memory with gate and state level fusion for light field-based face recognition. *IEEE Transactions on Information Forensics and Security*, *16*(04), 1365–1379. doi:10.1109/TIFS.2020.3036242

Tamy, S., Belhadaoui, H., Rabbah, M., Rabbah, N., & Rifi, M. (2019). Select the best machine learning algorithms for prediction and classification of intrusions using kdd99 In-De dataset. *Indian Journal of Science and Technology*, *12*(37), 485–551. doi:10.17485/ijst/2019/v12i37/147551

Tang, Z., Hu, H., & Xu, C. (2022). A federated learning method for network In-De. *Concurrency and Computation*, *34*(10), 138–146. doi:10.1002/cpe.6812

Yao, J., Xing, W., Wang, D., Xing, J., & Wang, L. (2021). Active dropblock: Method to enhance deep model accuracy and robustness. *Neurocomputing*, *454*(68), 101–109. doi:10.1016/j.neucom.2021.04.101

Yin, C. L., Zhu, Y. F., & Fei, J. L. (2020). A D-L approach for In-De using recurrent neural networks. *IEEE Access: Practical Innovations, Open Solutions*, (5), 21954–21961.

Zavrak, S., & Iskefiyeli, M. (2020). Anomaly-based In-De from network flow features using variational auto encoder. *IEEE Access: Practical Innovations, Open Solutions*, *8*(99), 108346–108358. doi:10.1109/ACCESS.2020.3001350

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on D-L for big data. *Information Fusion*, *42*(22), 146–157. doi:10.1016/j.inffus.2017.10.006

Zhang, X., Ji, J., Wang, L., He, Z., & Liu, S. (2021). People's fast moving detection method in buses based on yolov5. *International Journal of Sensors and Sensor Networks*, *9*(1), 15–43.