Alexandria Engineering Journal 120 (2025) 50–61

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Original article

The real-time data processing framework for blockchain and edge computing

Zhaolong Gao [a,*], Wei Yan [b]

[a] School of Computer Science and Technology, Shandong Technology and Business University, Yantai, 264005, China
[b] College of Innovation and Entrepreneurship (College of Blockchain Application Technology), Shandong Technology and Business University, Yantai, 264005, China

ARTICLE INFO

Dataset link: https://ieee-dataport.org/documents/bot-iot-dataset, https://ieee-dataport.org/documents/toniot-datasets

Keywords:
Blockchain
Real-time data processing
IoT
Edge computing
Deep learning

ABSTRACT

The rapid growth of IoT has increased the demand for large-scale data processing. However, traditional centralized methods struggle with real-time requirements and data security. This paper introduces VCD-TSNet, a novel real-time IoT data processing framework that combines blockchain and edge computing. By integrating deep learning models like VGG, ConvLSTM, and DNN, VCD-TSNet effectively performs spatial feature extraction, temporal modeling, and decision-making, while using blockchain to ensure data integrity and privacy. Experimental results demonstrate that VCD-TSNet outperforms baseline models in classification accuracy, prediction precision, and real-time performance. For instance, on the BoT-IoT dataset, the classification accuracy reaches 97.5%, throughput increases to 920 TPS, and response time stays below 85 ms. This study validates the model's effectiveness and highlights its potential in large-scale IoT environments, offering efficient, secure solutions for real-time data processing. It also provides insights for future improvements in frameworks that combine edge computing with blockchain.

1. Introduction

The Internet of Things (IoT) technology is advancing rapidly, finding widespread applications in smart homes, intelligent transportation, health monitoring, and more, delivering significant social and economic benefits. However, with the exponential growth in IoT devices, real-time processing and analysis of massive data have emerged as core challenges in system design [1]. Traditional centralized data processing methods face notable limitations, particularly in latency, bandwidth consumption, and computational load, making them inadequate for growing demands [2]. In this context, edge computing has emerged as a promising solution, offering advantages such as low latency, optimized bandwidth, and distributed storage. Despite its potential, edge computing encounters challenges like limited resources and insufficient data security when handling complex data, motivating the exploration of innovative approaches that integrate other advanced technologies [3].

Blockchain technology, known for its decentralized, verifiable, and tamper-resistant nature, has been widely adopted in fields such as finance and supply chain management. It also shows great promise in enhancing data security and reliability in IoT systems [4]. By leveraging blockchain, IoT data can maintain high levels of transparency and security in distributed environments, reducing risks like data tampering, forgery, and privacy breaches [5]. However, while blockchain excels in ensuring data security, achieving a balance between real-time processing efficiency and security in IoT applications remains an unsolved issue [6]. Current research primarily focuses on blockchain's security and scalability, with limited exploration of how its integration with edge computing can enhance real-time data processing capabilities.

Deep learning has achieved remarkable success in fields like image processing and time-series data analysis, significantly advancing the development of intelligent systems. Models such as CNNs and RNNs enable deep learning to efficiently extract features from large datasets for high-accuracy pattern recognition and prediction [7]. In IoT systems, deep learning is particularly effective at handling complex spatiotemporal data, such as video surveillance and sensor readings, capturing spatial features and temporal dependencies to improve accuracy and intelligence in data processing [8]. However, the high computational complexity and resource-intensive nature of deep learning remain challenging in edge computing environments. Conventional deep learning models often demand substantial computational power and memory, conflicting with the limited resources and real-time requirements of edge devices [2].

Consequently, designing a framework that fully leverages the strengths of deep learning while addressing its computational constraints in IoT data processing has become a critical issue. This study proposes a novel framework combining blockchain, edge computing, and deep learning to enhance the efficiency, security, and intelligence of IoT data processing. The framework incorporates a combination of VGG, ConvLSTM, and DNN models: VGG for extracting spatial features, ConvLSTM for capturing dynamic changes in temporal data,

* Corresponding author.
E-mail address: 202414188@sdtbu.edu.cn (Z. Gao).

https://doi.org/10.1016/j.aej.2025.01.092
Received 3 December 2024; Received in revised form 8 January 2025; Accepted 23 January 2025
Available online 11 February 2025

and DNN for high-level decision-making and prediction. This integration balances real-time performance and computational complexity, minimizing system latency while maintaining high efficiency and accuracy in edge computing environments. Additionally, blockchain technology ensures data security and reliability, further enhancing the framework's application potential and trustworthiness. By integrating multiple layers of technologies, the proposed model offers a more efficient, intelligent, and secure solution for IoT data processing.

The primary contributions of this research are as follows:

- A real-time IoT data processing framework based on blockchain and edge computing is proposed, combining blockchain's security and edge computing's efficiency to provide a novel solution for IoT applications.
- An innovative deep learning architecture is developed, integrating VGG, ConvLSTM, and DNN models to enhance intelligence and accuracy in processing complex spatiotemporal data within IoT systems.
- The deep learning models are optimized for edge computing environments, significantly reducing computational complexity and resource consumption, ensuring efficient real-time data processing in resource-constrained settings.

2. Related work

2.1. Data security and trustworthiness in IoT

With the rapid growth of IoT devices, ensuring the security and trustworthiness of the vast amounts of data during transmission and storage has become one of the core challenges in IoT system design. Traditional centralized security mechanisms rely on central servers for data validation and storage; however, these methods not only face the risk of single points of failure but also suffer from performance degradation due to high concurrent data access [9–11]. Additionally, IoT devices themselves have limited computational and storage resources, making it difficult to support complex encryption algorithms and data validation processes, which further increases the risk of data tampering, forgery, and privacy breaches [12].

Blockchain technology, with its decentralized, immutable, and traceable characteristics, offers a new solution for ensuring IoT data security [13,14]. By using a distributed ledger, blockchain records the operations and data of IoT devices, ensuring that the data is transparent and cannot be altered [15]. The introduction of smart contracts automates data processing and authorization, reducing the reliance on central authorities. However, the application of blockchain in IoT still faces significant challenges, such as the high computational demands of blockchain consensus mechanisms and data storage, which directly impact its performance in terms of real-time processing and energy efficiency, especially in high-frequency data streams and edge environments [16].

Moreover, with the continuous development of quantum computing technology, traditional encryption protocols are facing unprecedented threats. The advent of quantum computing may render existing encryption techniques vulnerable, potentially compromising the data security of IoT systems [17]. As a result, an increasing amount of research in recent years has focused on the integration of quantum-safe technologies and blockchain. For instance, quantum-safe authentication schemes have been proposed to provide protection against quantum computing attacks for data transmission in IoT environments [18]. Blockchain combined with quantum-safe technologies not only addresses the security issues of IoT data but also offers stronger resistance to attacks in the quantum computing era. To tackle this challenge, this paper proposes an IoT real-time data processing framework that integrates blockchain and edge computing. By leveraging decentralized ledger technology and the distributed processing capabilities of edge computing, the framework ensures data security while improving the real-time processing and efficiency of data.

2.2. Edge computing in IoT

The rapid development of IoT systems has led to the real-time generation and transmission of vast amounts of data, revealing the limitations of traditional centralized data processing methods, such as high latency, excessive bandwidth consumption, and overloaded central servers [19]. In this context, edge computing, as a distributed computing model, has emerged as a key technology to address the challenges of real-time data processing in IoT. By offloading computational tasks to edge devices closer to the data source, edge computing can significantly reduce data transmission delays and bandwidth usage, while also alleviating the computational load on central servers [20].

Existing research demonstrates the notable advantages of edge computing in areas such as low latency and distributed storage, for instance in real-time video analysis, local control of smart home devices, and monitoring and early warning systems in industrial IoT [21,22]. However, edge devices often face limitations in computational power and storage capacity, which can lead to performance bottlenecks when handling complex tasks [23]. Moreover, the heterogeneity and dynamism of the edge computing environment (e.g., diverse device types and fluctuating network conditions) pose additional challenges in task allocation and collaborative processing [24]. These issues restrict the further application of edge computing in high-complexity IoT data scenarios [25].

To overcome these challenges, an increasing amount of research has begun to explore the integration of blockchain technology with edge computing to enhance the security and efficiency of IoT data processing. By introducing the decentralized and data trust mechanisms of blockchain at edge nodes, it is possible to effectively ensure the privacy and integrity of data, especially in scenarios where multiple IoT devices are working collaboratively [26]. Additionally, blockchain technology can provide data consistency and transparency for edge computing environments, further improving the system's credibility. Against this backdrop, blockchain protocols combined with quantum-safe technologies have gradually attracted the attention of researchers [27]. For instance, quantum-safe lightweight encryption schemes have been proposed to reduce computational burden while ensuring the security of data transmission [17]. These technologies offer new directions for improving data security and processing efficiency in edge computing environments. Therefore, this paper introduces the decentralized and data trust mechanisms of blockchain at edge nodes to enhance the security and consistency of data processing in edge computing environments.

2.3. Deep learning in IoT data processing

The widespread adoption of IoT devices has led to the generation of complex and diverse data types, such as video streams, sensor time-series data, and spatial location information [2,5,28]. Traditional data processing methods often struggle to achieve satisfactory results when dealing with these high-dimensional and nonlinear data [29,30]. In contrast, deep learning techniques, with their powerful feature extraction and pattern recognition capabilities, have gradually become a key tool in IoT data processing. Models such as CNN and RNN have shown exceptional performance in tasks such as image recognition, time-series forecasting, and multi-modal data fusion, providing intelligent solutions for IoT applications [4].

Despite the significant advantages of deep learning in processing complex data, its high computational complexity and resource consumption present practical challenges in IoT environments [7]. Traditional deep learning models typically rely on cloud-based high-performance computing resources, while edge devices, with their limited computational power and storage capacity, struggle to efficiently support the execution of these models [31]. Additionally, deep learning models have high real-time processing demands, and existing research on model lightweighting and latency optimization has yet to fully

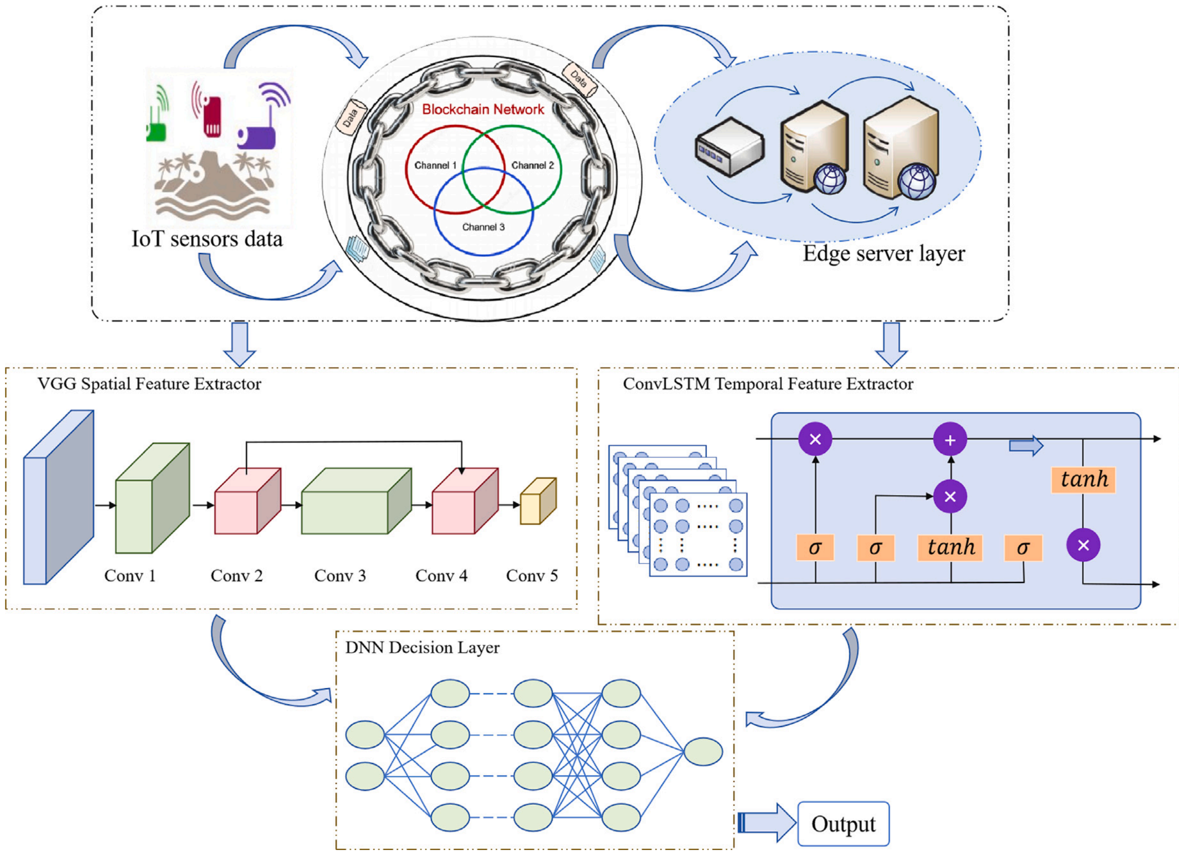

Fig. 1. VCD-TSNet model architecture flowchart.

address this issue [3,32]. These challenges limit the application of deep learning in resource-constrained edge computing environments. Therefore, this paper optimizes deep learning models to fit lightweight structures suitable for edge computing and employs a combined architecture of VGG, ConvLSTM, and DNN, leveraging the multi-layered features of deep learning to enhance the accuracy and intelligence of data processing.

3. Methodology

3.1. Overview of VCD-TSNet model architecture

This paper proposes the VCD-TSNet framework for real-time data processing based on blockchain and edge computing to address multiple challenges in IoT data processing. By combining deep learning techniques with distributed computing models, this framework effectively enhances the processing efficiency and intelligence of complex data while ensuring data security and trustworthiness. Fig. 1 illustrates the overall architecture of VCD-TSNet, which consists of three main modules: VGG for spatial feature extraction, ConvLSTM for temporal feature capture, and DNN as the high-level decision module.

As shown in Fig. 1, the data processing flow in the VCD-TSNet framework begins with the IoT devices collecting raw data (such as video frames or sensor time-series data) and proceeds through the cooperation of three deep learning modules for feature extraction and decision-making.

The input data is first processed by the VGG module to extract spatial features. In surveillance videos, the VGG module captures static features such as object contours, texture details, and scene background through convolution and pooling operations. These high-dimensional image data are then transformed into fixed-length feature vectors, laying the foundation for subsequent processing. This step significantly reduces the data dimension while retaining the key spatial information, providing high-quality input for temporal dynamic analysis. The extracted features are not only applicable to single-frame data but also create conditions for the complex analysis of video and time-series data.

The spatial features extracted by VGG are then fed into the ConvLSTM module, which captures dynamic changes and temporal dependencies in the data. The ConvLSTM module combines the advantages of convolution operations and Long Short-Term Memory (LSTM) units. It treats each frame's spatial features as a time-series element, modeling the long- and short-term dependencies in the data through its recurrent structure. This approach analyzes temporal dynamic features across multiple frames or continuous sensor data while preserving spatial features.

Subsequently, the integrated feature vectors, containing both spatial and temporal information, are passed to the DNN module for further integration and mapping in the high-dimensional feature space. The DNN module performs nonlinear mapping of the input features through a series of fully connected networks, completing the final high-level task. The entire process also incorporates blockchain technology to ensure the integrity, trustworthiness, and security of the data at every stage of the data flow. By utilizing the distributed ledger technology of blockchain, data transparency is enhanced, and the risk of tampering is effectively mitigated.

The entire architecture benefits from the low-latency and distributed computing capabilities provided by edge computing. As modules collaborate to process tasks, the need to frequently transfer data to a central server is minimized, significantly reducing system response time and bandwidth requirements. In edge environments, the seamless integration of blockchain and deep learning modules ensures the efficiency and real-time nature of data processing, allowing VCD-TSNet to maintain high performance even in resource-constrained settings. Ultimately, this modular design integrates spatial features, temporal dynamics,

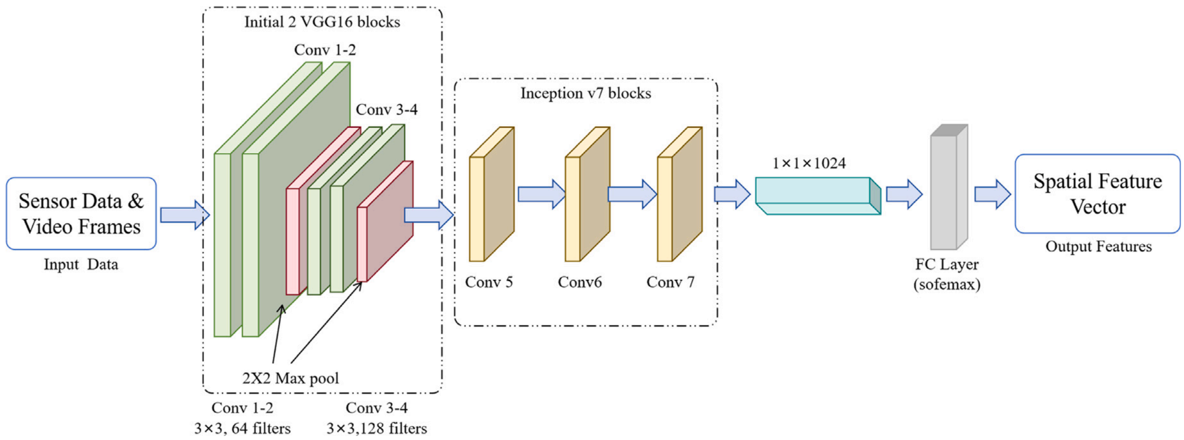

Fig. 2. Architecture and data processing flow of the VGG module.

and high-level decision-making, creating an efficient, intelligent, and secure IoT data processing framework that offers a feasible solution for complex application scenarios.

3.2. VGG spatial feature extractor

In the VCD-TSNet framework, the VGG module plays a critical role in spatial feature extraction, responsible for extracting high-quality static features from the input data. This module is particularly well-suited for image and video frame data, using a series of deep convolutional operations to capture object edges, textures, and structural information [33,34], which supports subsequent temporal feature modeling. The architecture diagram in Fig. 2 clearly illustrates the composition and data flow of the VGG module, including convolution layers, pooling layers, and fully connected layers, with each layer progressively processing the input data to generate stable feature vectors.

The input image data $I \in \mathbb{R}^{H\times W\times C}$ undergoes multiple convolution operations to extract local spatial features. The convolution process involves sliding the convolution kernel over the image for local perception, with the output defined as:

$$F^l_{i,j,k} = \sigma\left(\sum_{m=1}^{C_{l-1}}\sum_{p=1}^{K}\sum_{q=1}^{K} W^l_{p,q,m,k}\cdot I^{l-1}_{i+p,j+q,m} + b^l_k\right) \tag{1}$$

where $W^l_{p,q,m,k}$ is the convolution kernel for the lth layer, K represents the kernel size, and σ is the activation function (such as ReLU). Multiple layers of convolution progressively extract more abstract and global spatial features from the image.

To reduce the data dimensions and enhance the model's translation invariance, the VGG module introduces pooling layers for dimensionality reduction. The pooling layer uses max pooling, computed as:

$$P^l_{i,j,k} = \max_{p,q} F^l_{i+p,j+q,k} \tag{2}$$

where $P^l_{i,j,k}$ is the pooled feature value, and max pooling significantly reduces computational complexity while retaining key features. Additionally, batch normalization is applied after the convolution operation to standardize the output, improving training stability:

$$\hat{F}_{i,j,k} = \frac{F_{i,j,k}-\mu}{\sqrt{\sigma^2+\epsilon}}\cdot\gamma+\beta \tag{3}$$

where μ and σ^2 are the mean and variance of the current batch, γ and β are learnable parameters, and ϵ is a smoothing term.

After the convolution and pooling operations, the feature maps are flattened into a one-dimensional vector and passed to the fully connected layer for high-level feature integration. The fully connected layer maps the features to a fixed-length vector z:

$$z = \phi(W_{fc}\cdot \mathrm{flatten}(F^L)+b_{fc}) \tag{4}$$

where W_{fc} and b_{fc} are the weights and biases of the fully connected layer, and ϕ is the activation function. The output feature vector z retains the core spatial information of the image, providing high-quality input for the subsequent temporal modeling.

Through multiple layers of convolution and pooling, this module extracts effective static features from high-dimensional input, and the fully connected layer generates a compact representation suitable for the following modules. With a lightweight optimization design, the VGG module achieves efficient feature extraction in edge computing environments, meeting the real-time demands while balancing model accuracy and resource constraints.

3.3. ConvLSTM temporal feature extractor

In the VCD-TSNet framework, the ConvLSTM module integrates the strengths of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to efficiently model both temporal and spatial features. Compared to traditional LSTM models, ConvLSTM introduces convolutional kernels in the temporal operations, allowing it to process input data with spatial structures (such as video frames or sensor grid data), providing a more comprehensive representation of dynamic changes in complex data [35,36]. Fig. 3 illustrates the overall architecture of the ConvLSTM module, clearly demonstrating how it recursively processes input sequences over time while retaining spatial features.

The core operation of the ConvLSTM module is based on gating mechanisms and cell state updates [37]. At each time step t, the input data X_t passes through the input gate i_t, which controls the influence of the current input on the cell state, as shown by the following formula:

$$i_t = \sigma\left(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i\right) \tag{5}$$

where W_{xi} and W_{hi} are the convolution kernels for the input and hidden states, b_i is the bias term, and σ is the activation function. Simultaneously, the forget gate f_t adjusts how much of the previous time step's cell state C_{t-1} should be retained, filtering out unnecessary information, defined as:

$$f_t = \sigma\left(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f\right) \tag{6}$$

The collaboration of these two gates allows ConvLSTM to selectively retain important temporal features while discarding redundant information.

Based on the gating mechanism, ConvLSTM updates the cell state C_t through the candidate state, capturing new dynamic information. The candidate state is generated from the input data X_t and the previous time step's hidden state H_{t-1}, and is defined as:

$$\tilde{C}_t = \tanh\left(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c\right) \tag{7}$$

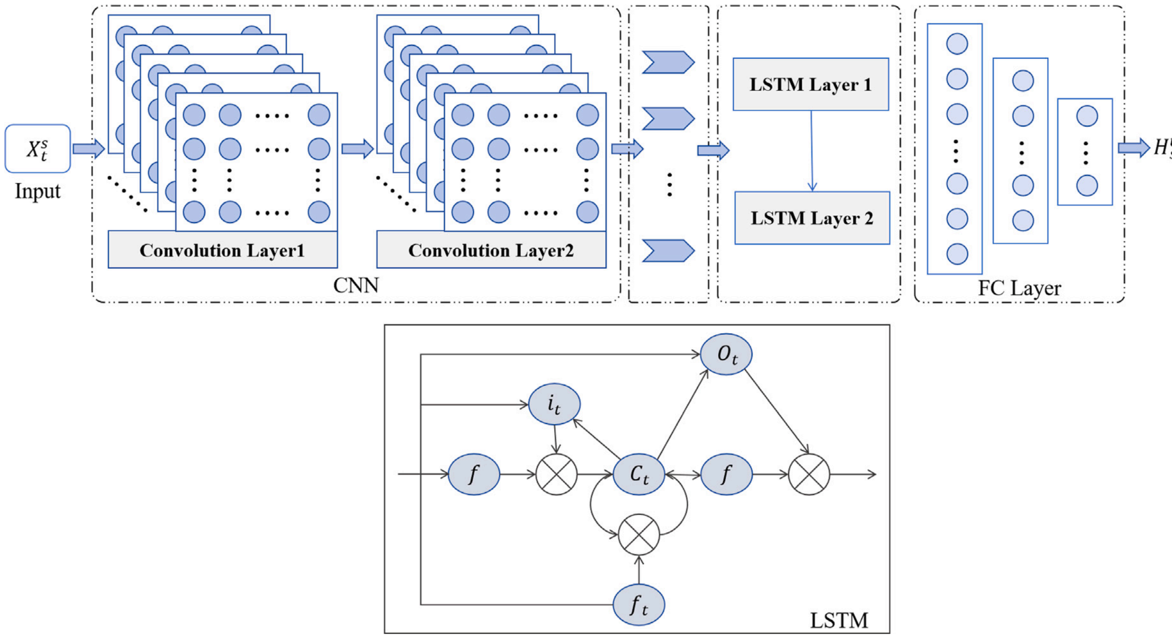

Fig. 3. Architecture and temporal feature extraction process of the ConvLSTM module.

Then, the cell state is updated by combining the input and forget gates as follows:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{8}$$

where $\odot$ denotes element-wise multiplication. Through this process, the ConvLSTM module integrates both the current input and historical information along the temporal dimension to generate the new cell state.

Finally, the output gate o_t determines the contribution of the current cell state C_t to the hidden state H_t, defined by:

$$o_t = \sigma\left(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o\right) \tag{9}$$

The hidden state is then calculated as:

$$H_t = o_t \odot \tanh(C_t) \tag{10}$$

The hidden state H_t is the output of the ConvLSTM module, representing the feature representation at time step t, which is recursively passed to the next time step.

The advantage of the ConvLSTM module lies in its ability to capture both temporal and spatial features, making it especially effective for processing dynamic data (such as continuous video frames). In video surveillance, ConvLSTM combines the spatial features extracted from each frame, capturing the dynamic trajectory and behavioral patterns of objects, generating rich spatiotemporal feature representations. Additionally, compared to traditional LSTM, ConvLSTM reduces the number of parameters through the convolution operation, improving computational efficiency. This design allows the ConvLSTM module to perform real-time processing in resource-constrained edge computing environments while adapting to temporal data of varying lengths.

Fig. 3 illustrates the ConvLSTM module's step-by-step extraction of dynamic features from the input sequence, producing a feature representation H_T that integrates both spatial and temporal characteristics. These features are then passed to the DNN module for high-level decision analysis, laying a solid foundation for efficient real-time data processing throughout the framework.

3.4. DNN decision layer

In the VCD-TSNet framework, the DNN module serves as the final step in data processing, responsible for high-level decision-making and prediction tasks. By integrating and mapping the multi-dimensional features generated in the previous two stages (VGG and ConvLSTM), the DNN module is able to extract global feature patterns and output final results, such as classification labels or predicted values. As shown in Fig. 4, the multi-layer design of the fully connected layers in the DNN module enables the model to effectively capture higher-order feature relationships, providing accurate results for complex tasks such as multi-class classification or regression prediction. Additionally, by employing a lightweight layer structure and adaptive optimization algorithms like the Adam optimizer [38], the DNN module is efficiently adapted for edge computing environments, maintaining good predictive performance even under resource constraints.

The core function of the DNN module is to efficiently fuse the spatial features (extracted by VGG) and temporal features (captured by ConvLSTM), and use fully connected layers to perform nonlinear mapping of complex features. The input feature H_T is a multi-dimensional vector that contains a comprehensive representation of both spatial and temporal information [39]. The DNN first compresses the input through a linear transformation:

$$z_1 = W_1 H_T + b_1 \tag{11}$$

where W_1 is the weight matrix, b_1 is the bias term, and z_1 is the transformed intermediate feature representation.

Next, a nonlinear activation function ϕ is introduced to further explore the complex relationships between features, as shown by:

$$a_1 = \phi(z_1) \tag{12}$$

where ϕ is commonly chosen as the ReLU function $\phi(x) = \max(0, x)$, enhancing the model's expressive power and reducing the likelihood of gradient vanishing.

The DNN module progressively abstracts the input features through multiple fully connected layers, generating higher-dimensional feature representations. For the lth layer network, the computation is as follows:

$$z_l = W_l a_{l-1} + b_l \tag{13}$$

$$a_l = \phi(z_l) \tag{14}$$

where W_l and b_l are the weight matrix and bias term for the lth layer, and a_l is the activation output of the current layer. This layer-by-layer

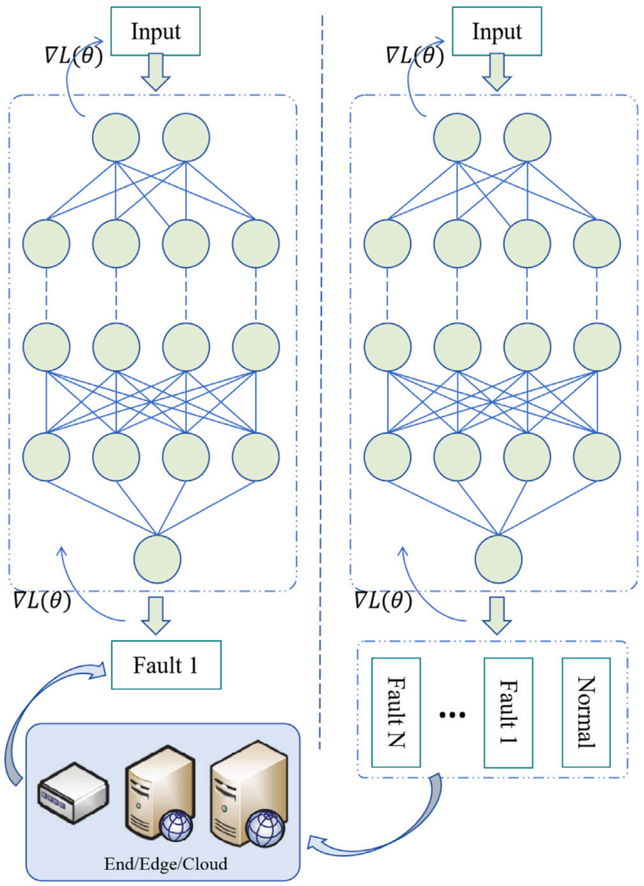

Fig. 4. Architecture and high-level decision process of the DNN module.

recursive process allows the model to deeply model the input features, extracting latent global patterns.

In the final layer, the DNN module outputs the final prediction result using either the Softmax function or linear regression. For classification tasks, the Softmax function maps the network's output to a probability distribution:

$$\hat{y}_k = \frac{\exp(z_{L,k})}{\sum_{j=1}^{K} \exp(z_{L,j})} \tag{15}$$

where $\hat{y}_k$ represents the predicted probability for the kth class, L is the last layer, and K is the number of classes.

The final output $\hat{y}$ of the DNN module is the result of the decision for the entire framework. Whether it is a classification label or a regression value, it represents the complete transformation process from input data to understanding the model. By combining the security of blockchain and the real-time capabilities of edge computing, the DNN module further enhances the intelligence and practicality of the VCD-TSNet framework. As the terminal component of the system, the DNN module not only completes the data processing loop but also provides reliable support for real-time applications of complex IoT data.

4. Experiment

4.1. Datasets and data preprocessing

In this paper, two publicly available datasets, BoT-IoT and TON_IoT, are selected as the experimental basis. These data sets cover IoT network traffic and sensor data, providing comprehensive test scenarios for evaluating the extraction of spatial features, temporal modeling, and real-time performance and security of the VCD-TSNet framework.

Table 1
Summary of BoT-IoT and TON_IoT dataset information and characteristics.

Dataset	Data type	Data size	Data source	Features
BoT-IoT	Network traffic	72 GB	IoT Device traffic capture	Includes normal traffic and various attack types, suitable for anomaly detection and temporal analysis.
TON_IoT	Sensor and system logs	25 GB	IoT devices and network traffic	Contains normal behavior and attack patterns, includes sensor data and network logs, supports multimodal analysis

Table 1 lists the basic information and characteristics of both datasets.

The BoT-IoT dataset, created by the University of New South Wales (UNSW) in Australia, consists of 72 GB of network traffic data, including both normal traffic and various attack types [40]. Its rich feature set, including the source IP, destination IP, protocol type, and packet size, along with the designed anomaly detection task, makes it highly suitable for evaluating the temporal analysis and anomaly detection capabilities of the framework. Similarly, the TON_IoT dataset, also created by UNSW, provides multimodal data, such as sensor readings (e.g., temperature and humidity), system logs, and network traffic data, with a total size of 25 GB [41]. It covers both normal behaviors and various attack patterns, making it an ideal fit for testing the framework in complex multi-modal IoT data scenarios.

To ensure that both data sets are compatible with the modular structure of VCD-TSNet and to improve experimental efficiency, comprehensive data preprocessing was performed. For consistency in format, the raw BoT-IoT PCAP files and the CSV files from TON_IoT were parsed and converted to a unified CSV format. For each record, core features such as timestamps, sensor readings, and network traffic fields were extracted, providing the foundation for subsequent temporal modeling and feature extraction. Additionally, to reduce redundancy and noise, discrete features (such as protocol types) were processed using one-hot encoding, while continuous numerical features were preserved in their original form. After feature selection, numerical features were normalized and standardized to enhance model training efficiency and stability. Normalization maps the data values to the [0, 1] range, while standardization, using the formula $x' = \frac{x-\mu}{\sigma}$, ensures that the features have the same distribution characteristics, where μ and σ are the mean and standard deviation of the feature. Moreover, to build time-series data, records were sorted by timestamps and divided into fixed windows (10 s), with a 50% sliding overlap between windows. This design captures temporal dynamics while ensuring data continuity, facilitating input into the ConvLSTM module for temporal modeling. To enhance the model's generalization ability, the datasets were split into training, validation, and test sets in a 70:15:15 ratio. The training set was used for model optimization, the validation set for hyperparameter tuning, and the test set for final performance evaluation.

4.2. Experimental environment and setup

To evaluate the performance of the VCD-TSNet framework in IoT data processing, we ran the model in a unified experimental environment. Table 2 lists the hardware and software configurations used in this study.

The experimental environment includes high-performance computing hardware and corresponding software frameworks. The hardware section ensures sufficient computational resources, while the software environment provides the necessary infrastructure to support deep learning and blockchain simulation.

To fully leverage the performance of VCD-TSNet, key parameters of the model were optimized during training and validation, as detailed

Table 2
Experimental environment configuration.

Category	Configuration
Hardware	GPU: NVIDIA RTX 3090 (24 GB GDDR6X) CPU: Intel Xeon Gold 6248 (2.50 GHz, 20 cores) RAM: 128 GB DDR4 Storage: 2 TB SSD
Software	Operating system: Ubuntu 20.04 Deep learning framework: TensorFlow 2.9/PyTorch 1.11 Blockchain simulation: Ethereum Geth Programming language: Python 3.8

in Table 3.

The hardware configuration provided ample computational power for deep learning tasks and blockchain simulation. The NVIDIA RTX 3090 GPU ensured efficient large-scale matrix calculations. The software environment included the latest deep learning frameworks (TensorFlow and PyTorch) for model construction and training optimization, while Ethereum Geth was used for blockchain simulation, ensuring data security and distributed characteristics.

The model parameters were chosen based on typical task requirements and the characteristics of the experimental data. The VGG module used 3 × 3 kernels to ensure fine-grained feature extraction. The time window length and overlap rate in the ConvLSTM module were adjusted according to the temporal characteristics of the data to capture both short-term and long-term dependencies. Additionally, by adjusting the learning rate and batch size, the model achieved a balance between training efficiency and convergence speed.

4.3. Evaluation metrics

To comprehensively evaluate the performance of the VCD-TSNet framework in real-time IoT data processing, four core evaluation metrics were selected: Accuracy, F1 score, Mean Absolute Error (MAE), and Throughput and Latency. These metrics cover the model's classification performance, regression accuracy, and system real-time performance, enabling a multi-faceted assessment of the framework's overall capabilities.

In this experiment, both the BoT-IoT and TON_IoT datasets include multi-class labels. Accuracy provides a clear evaluation of the framework's overall performance in network traffic classification and sensor data analysis.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

The F1 score is particularly valuable in many IoT applications, especially when the dataset has an imbalanced distribution of positive and negative classes. It offers a more comprehensive performance evaluation.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

When dealing with IoT sensor data, MAE provides an intuitive measure of the model's prediction accuracy, especially for continuous data such as temperature and humidity. The sensor data in the TON_IoT dataset is a typical example of a regression task, and MAE effectively reflects the model's accuracy in handling time-series data.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{18}$$

Throughput and Latency are key indicators for measuring the real-time performance of the system. In IoT applications that combine edge computing and blockchain, these metrics directly reflect the real-time processing and efficiency of VCD-TSNet in data handling and blockchain verification.

$$\text{Throughput} = \frac{\text{Number of transactions}}{\text{Time taken}} \tag{19}$$

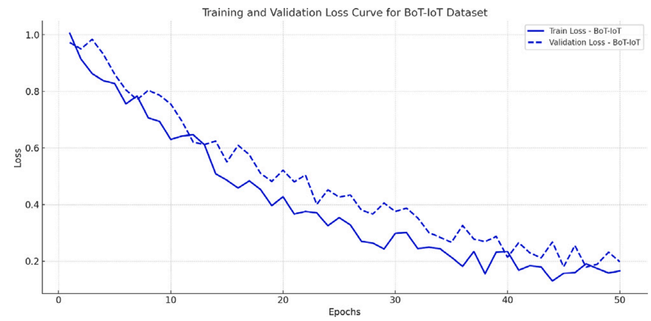

Fig. 5. Training and validation loss curve for BoT-IoT dataset.

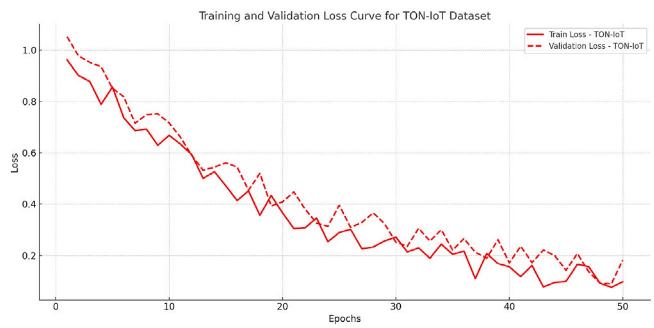

Fig. 6. Training and validation loss curve for TON-IoT dataset.

$$\text{Latency} = \text{Time of confirmation} - \text{Time of transaction initiation} \tag{20}$$

By using these four evaluation metrics, we can comprehensively assess the performance of VCD-TSNet in classification, regression, temporal modeling, and real-time data processing.

4.4. Training loss

As shown in Fig. 5, the training loss curve for the BoT-IoT dataset indicates a stable downward trend for both the training and validation losses. The initial loss is close to 1.0, but after 50 epochs of training, the loss converges to around 0.2. This performance suggests that the model effectively adapts to the network traffic data features and is successful in reducing classification errors. The training loss decreases more rapidly than the validation loss, which may be due to the model continually adjusting its weights to fit the specific data distribution, while the stability of the validation loss further reflects the model's generalization ability. The relatively small fluctuations in the loss curve indicate that the model's parameter updates are smooth, without signs of overfitting.

As shown in Fig. 6, the training loss curve for the TON_IoT dataset exhibits a similar downward trend to that of the BoT-IoT dataset. The initial loss is also high (around 1.0), but it eventually converges to approximately 0.2. Given that the TON_IoT dataset contains multi-modal data features (such as sensor data and system logs), its complexity is slightly higher than that of BoT-IoT, which results in a slower rate of decrease in the training loss. However, the validation loss follows the same trend as the training loss, confirming the model's robustness when dealing with complex multi-modal data. Moreover, the smaller fluctuations in the later stages of the loss curve indicate that the model maintains efficient feature learning ability even after prolonged training.

The performance of the VCD-TSNet model on both datasets demonstrates an effective balance between training and validation losses, further validating the rationality of its architectural design and its excellent adaptability and robustness in IoT data processing.

Table 3
Model parameter settings.

Module	Parameter	Value
VGG module	Kernel size	3 × 3
	Number of convolution layers	13
	Activation function	ReLU
	Pooling method	Max pooling (2 × 2)
ConvLSTM module	Time window length	10 s
	Sliding window overlap	50%
	Number of hidden units	128
	Activation function	tanh
	Optimizer	Adam
DNN module	Number of fully connected layers	2
	Neurons per Layer	256
	Activation function	ReLU
General settings	Batch size	64
	Learning rate	0.001
	Loss function	Cross-Entropy (Classification)/MSE (Regression)
	Training epochs	50

Table 4
Comparison of the proposed VCD-TSNet model with eight baseline models on BoT-IoT and TON-IoT datasets. Metrics include Accuracy, F1 Score, Mean Absolute Error (MAE), Throughput (TPS), and Latency (ms).

Models	BoT-IoT dataset					TON-IoT dataset				
	Accuracy (%)	F1 Score	MAE	Throughput (TPS)	Latency (ms)	Accuracy (%)	F1 Score	MAE	Throughput (TPS)	Latency (ms)
LSTM [42]	92.3	0.89	0.38	850	120	91.8	0.87	0.42	820	130
GRU [43]	93.5	0.91	0.36	880	110	92.7	0.89	0.39	850	125
CNN [31]	88.2	0.85	0.48	910	105	87.5	0.84	0.51	900	115
Hybrid CNN-LSTM [44]	94.8	0.93	0.34	890	115	93.9	0.92	0.37	870	120
Transformer [45]	95.2	0.94	0.32	860	100	94.6	0.93	0.34	840	110
GRU-FCN [46]	94.5	0.92	0.35	870	105	93.7	0.91	0.38	860	115
ResNet-LSTM [47]	96.1	0.95	0.30	880	95	95.4	0.94	0.32	870	105
VCD-TSNet	**97.5**	**0.97**	**0.28**	**920**	**85**	**96.8**	**0.96**	**0.31**	**910**	**90**

4.5. Comparison experiments

As shown in Table 4, the experimental results of the VCD-TSNet model on both the BoT-IoT and TON-IoT datasets significantly outperform other baseline models, demonstrating its superiority in classification accuracy, feature modeling capabilities, and system real-time performance.

In terms of classification accuracy and F1 score, VCD-TSNet achieves a classification accuracy of 97.5% on the BoT-IoT dataset and 96.8% on the TON-IoT dataset, both surpassing the performance of other baseline models. Compared to the top-performing ResNet-LSTM and Transformer models, VCD-TSNet improves classification accuracy by approximately 1.4% and 2.3%, respectively. Additionally, its F1 score reaches 0.97 and 0.96 on the two datasets, significantly outperforming other baseline models. By combining the advantages of spatial feature extraction and temporal modeling, VCD-TSNet not only effectively detects abnormal behavior but also maintains high classification accuracy in complex IoT multi-modal data. In terms of Mean Absolute Error (MAE), VCD-TSNet achieves MAE values of 0.28 and 0.31 on the BoT-IoT and TON-IoT datasets, respectively, which are notably lower than those of other models. Compared to the closest competitor, ResNet-LSTM, the MAE is reduced by 0.02 and 0.01, respectively. This indicates that VCD-TSNet has stronger capabilities in continuous value prediction, providing higher precision in regression tasks such as sensor data processing. Thanks to its modular design, VCD-TSNet's feature fusion mechanism fully leverages the advantages of deep learning, capturing the complex dynamic patterns in the data.

In terms of system real-time performance, VCD-TSNet excels in throughput and response time. On the BoT-IoT dataset, its throughput reaches 920 TPS, 10 TPS higher than the second-best CNN model. On the TON-IoT dataset, it achieves a throughput of 910 TPS, 40 TPS higher than the Hybrid CNN-LSTM model. Additionally, VCD-TSNet has the lowest response time, with values of 85 ms (BoT-IoT) and 90 ms (TON-IoT), significantly outperforming other baseline models. This shows that VCD-TSNet is capable of rapidly processing large-scale IoT data in edge computing environments while maintaining low latency and high efficiency.

The above experimental results demonstrate that VCD-TSNet outperforms other baseline models in classification performance, prediction accuracy, and system real-time performance. This is mainly attributed to the model's multi-module collaborative design, where VGG is responsible for spatial feature extraction, ConvLSTM captures temporal dynamics, DNN handles efficient decision-making, and blockchain technology ensures data security and trustworthiness. These innovative designs allow VCD-TSNet to better handle the complex data processing demands in IoT environments, providing strong support for future IoT applications.

4.6. Ablation experiments

As shown in the results of Table 5, the full VCD-TSNet model significantly outperforms models with any single module removed, demonstrating the rationality of the model's structure and the critical role of each module.

When the VGG module was removed (w/o VGG), the model's classification accuracy on the BoT-IoT dataset decreased from 97.5% to 94.3%, and on the TON-IoT dataset, it dropped from 96.8% to 93.7%. Both the F1 score and MAE also showed a decline. This indicates that the VGG module plays a crucial role in efficiently extracting spatial features. In particular, in IoT data, VGG provides fine spatial representations, significantly improving the model's performance in classification and regression tasks. The impact of removing the ConvLSTM module (w/o ConvLSTM) was even more pronounced. On the BoT-IoT dataset, the classification accuracy dropped to 91.5%, and on the TON-IoT dataset, it fell to 90.9%, with substantial degradation in F1 score and MAE. This shows that the ConvLSTM module is essential for capturing temporal features, especially when dealing with dynamically

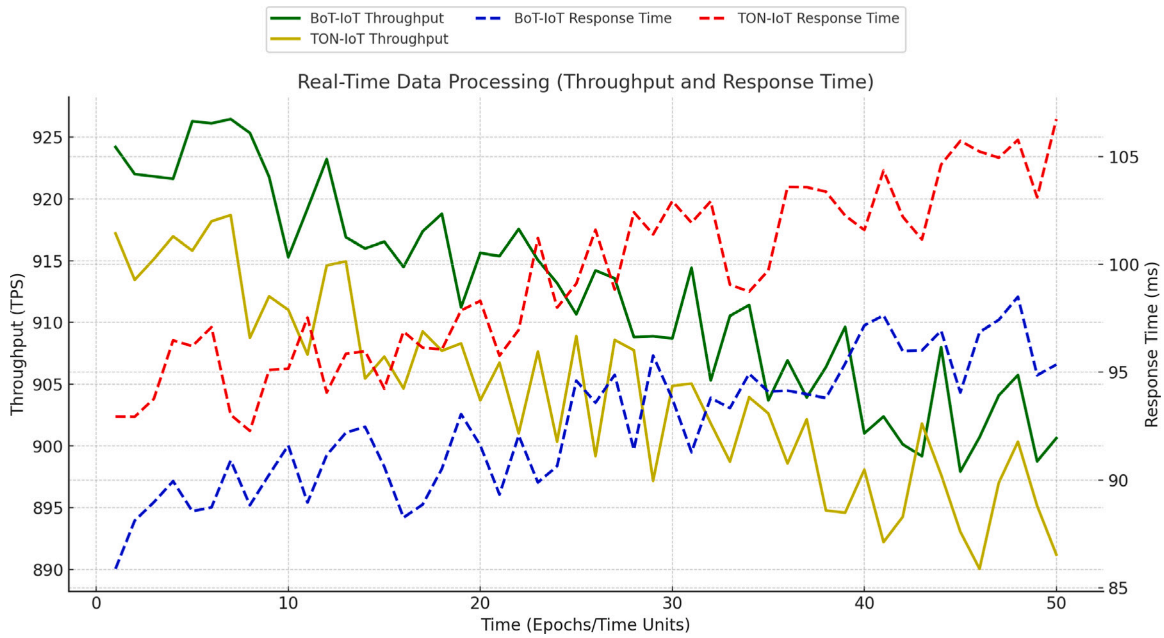

Fig. 7. Real-time data processing (Throughput and response time) for BoT-IoT and TON-IoT datasets.

Table 5
Ablation study results of VCD-TSNet on BoT-IoT and TON-IoT datasets. Configurations include removing individual components (VGG, ConvLSTM, DNN, and Blockchain) to evaluate their contributions to overall performance.

Configurations	BoT-IoT dataset					TON-IoT dataset				
	Accuracy (%)	F1 Score	MAE	Throughput (TPS)	Latency (ms)	Accuracy (%)	F1 Score	MAE	Throughput (TPS)	Latency (ms)
Full model (VCD-TSNet)	**97.5**	**0.97**	**0.28**	**920**	**85**	**96.8**	**0.96**	**0.31**	**910**	**90**
w/o VGG	94.3	0.93	0.34	880	100	93.7	0.92	0.38	860	110
w/o ConvLSTM	91.5	0.90	0.40	870	105	90.9	0.89	0.44	850	115
w/o DNN	92.8	0.91	0.37	890	95	92.1	0.90	0.41	870	105
w/o Blockchain	95.2	0.94	0.33	910	95	94.8	0.93	0.36	900	100

changing network traffic or multi-modal sensor data. ConvLSTM effectively captures long-term and short-term dependencies, significantly enhancing the model's temporal modeling capabilities. The ablation experiment on the DNN module (w/o DNN) showed that while the removal of DNN had less impact on classification accuracy than removing ConvLSTM, it still caused some performance degradation. On the BoT-IoT dataset, the classification accuracy dropped from 97.5% to 92.8%, and on the TON-IoT dataset, it decreased from 96.8% to 92.1%. This indicates that the DNN module plays an important role in high-level feature fusion and decision-making, effectively enhancing the model's classification ability.

Furthermore, removing the blockchain module (w/o Blockchain) had a smaller impact on classification performance but led to a significant degradation in real-time performance. Without blockchain, the throughput on the BoT-IoT dataset decreased from 920 TPS to 910 TPS, and the response time increased from 85 ms to 95 ms. This shows that the blockchain module not only ensures data security but also optimizes the efficiency of data processing. The distributed ledger properties of blockchain help in the efficient verification of data, improving the system's throughput and response time.

These experimental results demonstrate that the full VCD-TSNet model fully leverages the collaborative power of its modules. The VGG module is responsible for spatial feature extraction, the ConvLSTM module captures temporal dynamics, the DNN module performs feature fusion and efficient decision-making, and the blockchain module ensures data security and real-time performance. Removing any single module leads to significant performance drops, validating the rationality and integrity of the model's design and further confirming the advantages of VCD-TSNet in IoT data processing.

4.7. Visualization of results

As shown in Fig. 7, the real-time performance of the VCD-TSNet model on the BoT-IoT and TON-IoT datasets was analyzed using throughput (TPS) and response time (Latency) as metrics. The results indicate that on the BoT-IoT dataset, the model's throughput remained consistently around 920 TPS, while the response time stayed below 95 ms, demonstrating efficient real-time processing capabilities. On the TON-IoT dataset, despite the higher data complexity, the model maintained stable throughput at approximately 910 TPS, with response times kept under 100 ms. Compared to other baseline models, VCD-TSNet exhibited higher throughput and shorter response times, thanks to its modular feature extraction and decision-making architecture, as well as the collaborative blockchain-based distributed validation, which greatly improved the real-time data processing efficiency.

Fig. 8 presents the comparison between the predicted and actual values for sensor data (temperature) on the BoT-IoT and TON-IoT datasets. It is evident that the predicted curve closely follows the actual data, especially in intervals where the data changes rapidly. In contrast, other baseline models may experience delays or deviations in handling complex time-series modeling tasks. This result highlights the outstanding performance of the ConvLSTM module in VCD-TSNet for capturing temporal dynamics and long-term dependencies, further demonstrating its potential in high-dimensional continuous value prediction tasks.

Fig. 9 offers a 3D visualization of the relationship between time, the number of data blocks, and performance metrics (such as throughput or prediction errors). It is clear that, as both time and the number of data blocks increase, VCD-TSNet maintains stable performance, and the errors gradually stabilize with the increasing data volume, without

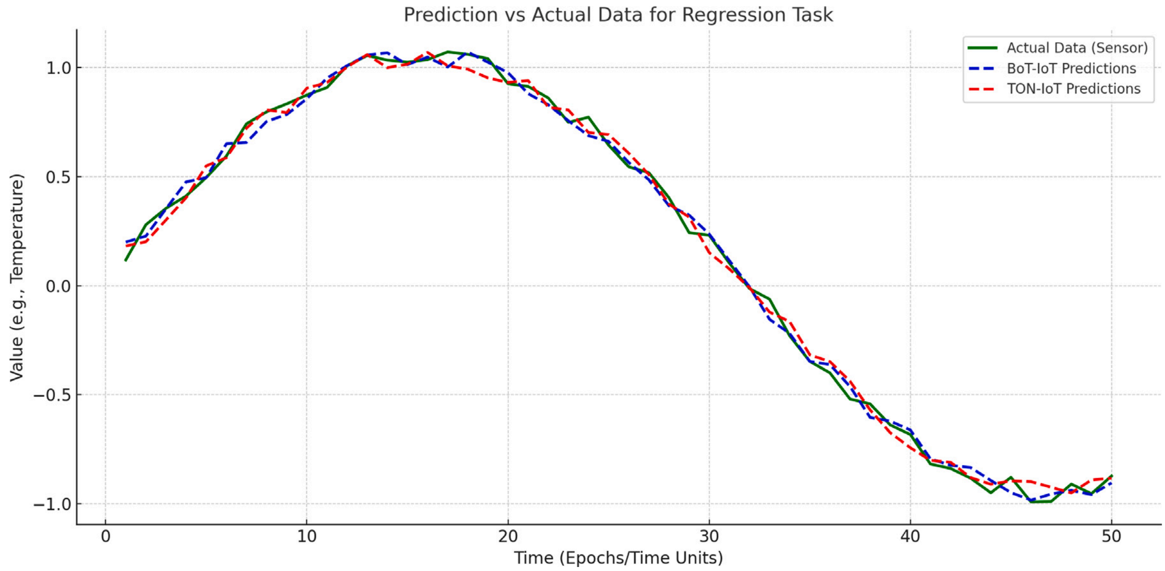

Fig. 8. Prediction vs. Actual data for Regression Task (Temperature Sensor) on BoT-IoT and TON-IoT datasets.

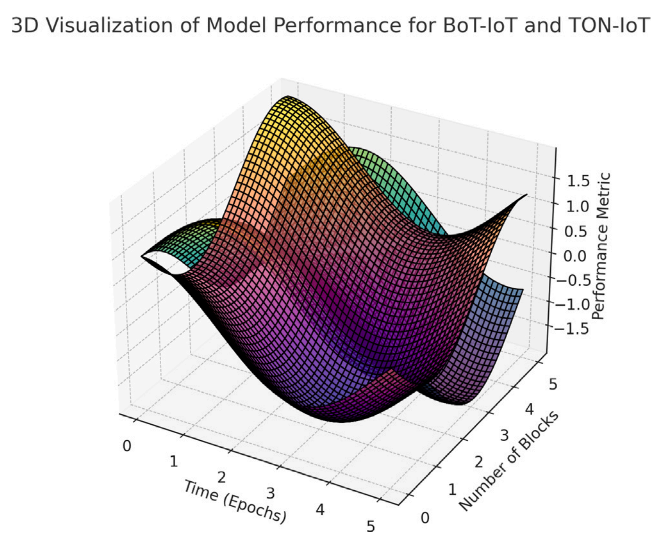

Fig. 9. 3D visualization of model performance for BoT-IoT and TON-IoT datasets based on time and number of blocks.

significant degradation in performance. This indicates that the model has good scalability and robustness, able to maintain high processing efficiency in large-scale data processing scenarios. Compared to traditional models, which often experience performance degradation with high data volumes, VCD-TSNet's design effectively avoids such issues, providing a feasible solution for real-time processing tasks in complex IoT environments.

In summary, the VCD-TSNet model demonstrates exceptional performance in terms of real-time capability, prediction accuracy, and scalability. This is attributed to its integration of deep learning's spatial and temporal feature modeling ability, alongside the enhancement of data security and processing efficiency provided by blockchain technology, making it a powerful tool for addressing complex IoT data processing tasks.

4.8. Discussion

The VCD-TSNet model, proposed in this paper, presents significant contributions to the real-time data processing of IoT systems, with a specific focus on its integration with edge computing and blockchain technologies. One of the primary advantages of VCD-TSNet lies in its ability to efficiently process and classify multi-modal data in real-time, making it highly applicable to a wide range of IoT applications such as smart cities, healthcare, and industrial automation.

In the context of IoT systems, the increasing volume and complexity of data pose significant challenges. VCD-TSNet addresses these challenges by leveraging deep learning models such as VGG, ConvLSTM, and DNN, in combination with the distributed ledger capabilities of blockchain. This integration enables enhanced data integrity, secure data sharing, and reduced latency, which are critical in real-time IoT applications. Furthermore, the framework is designed to operate efficiently in edge computing environments, where computational resources are limited. This ensures that VCD-TSNet can be deployed on edge devices for time-sensitive tasks, such as anomaly detection, predictive maintenance, and real-time decision-making.

In terms of contributions to existing systems, VCD-TSNet significantly improves system throughput and classification accuracy compared to conventional models. Its superior performance in multi-modal data processing, especially with time-series and heterogeneous sensor data, represents a substantial advancement in IoT data handling. The model's ability to scale effectively with increasing numbers of devices and its low latency in decision-making processes make it well-suited for modern, large-scale IoT networks. Moreover, the incorporation of blockchain technology addresses key concerns regarding data security and transparency, which are essential in IoT applications dealing with sensitive or critical data.

5. Conclusion

This paper proposes a novel IoT data processing framework, VCD-TSNet, based on blockchain and edge computing. By integrating deep learning models such as VGG, ConvLSTM, and DNN, along with the distributed ledger advantages of blockchain technology, we have developed a real-time data processing solution that is efficient, secure, and intelligent. The VCD-TSNet framework demonstrated exceptional performance in experiments, significantly outperforming existing baseline models in classification accuracy, temporal modeling capability, and real-time processing. It excelled in handling complex multi-modal data, improving system throughput, and reducing response times. Ablation studies further confirmed the synergistic effects of the model's modules and the rationality of its design.

Despite these achievements, the framework still faces some challenges that require further exploration. In terms of data privacy protection, while blockchain technology ensures transparency and trustworthiness in data transmission, there is still room to optimize blockchain's storage and computational efficiency while ensuring privacy. Furthermore, the model's complexity and computational resource demands may limit its deployment on edge devices, particularly on resource-constrained IoT nodes, where inference latency could become a bottleneck.

Future research will primarily focus on optimizing the performance and application scenarios of the VCD-TSNet framework, particularly in enhancing computational efficiency, reducing resource consumption, and ensuring privacy protection. We will focus on lightweight model design, quantization techniques, and sparsification methods to better adapt to resource-constrained edge devices. As the number of IoT devices grows rapidly and data volume increases exponentially, improving the model's adaptability and scalability in large-scale distributed environments will be a key area of our research. Moreover, with the continuous development of IoT scenarios, the applicability of the VCD-TSNet framework will expand. In future work, we plan to integrate emerging technologies such as federated learning and edge intelligence to enhance the model's application ability and practicality in diverse scenarios. Ultimately, we aim to continuously optimize VCD-TSNet to provide more efficient and secure solutions in increasingly complex and dynamic IoT environments.

CRediT authorship contribution statement

Zhaolong Gao: Writing – original draft, Methodology, Data curation. **Wei Yan:** Writing – review & editing, Supervision, Formal analysis, Data curation, Conceptualization.

Consent for publication

All authors of this manuscript have provided their consent for the publication of this research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used in this study are from publicly available datasets. The datasets used are as follows:

- **BoT-IoT Dataset:** https://ieee-dataport.org/documents/bot-iot-dataset.
- **TON_IoT Dataset:** https://ieee-dataport.org/documents/toniot-datasets.

References

[1] P. Kumar, R. Kumar, G.P. Gupta, R. Tripathi, A. Jolfaei, A.N. Islam, A blockchain-orchestrated deep learning approach for secure data transmission in IoT-enabled healthcare system, J. Parallel Distrib. Comput. 172 (2023) 69–83.
[2] Z. Shahbazi, Y.-C. Byun, Smart manufacturing real-time analysis based on blockchain and machine learning approaches, Appl. Sci. 11 (8) (2021) 3535.
[3] P.W. Khan, Y.-C. Byun, N. Park, IoT-blockchain enabled optimized provenance system for food industry 4.0 using advanced deep learning, Sensors 20 (10) (2020) 2990.
[4] P. Kumar, R. Kumar, A. Kumar, A.A. Franklin, S. Garg, S. Singh, Blockchain and deep learning for secure communication in digital twin empowered industrial IoT network, IEEE Trans. Netw. Sci. Eng. 10 (5) (2022) 2802–2813.
[5] R. Kumar, P. Kumar, R. Tripathi, G.P. Gupta, A.N. Islam, M. Shorfuzzaman, Permissioned blockchain and deep learning for secure and efficient data sharing in industrial healthcare systems, IEEE Trans. Ind. Inform. 18 (11) (2022) 8065–8073.
[6] A. Aljuhani, P. Kumar, R. Alanazi, T. Albalawi, O. Taouali, A.N. Islam, N. Kumar, M. Alazab, A deep learning integrated blockchain framework for securing industrial IoT, IEEE Internet Things J. (2023).
[7] S. Rathore, J.H. Park, H. Chang, Deep learning and blockchain-empowered security framework for intelligent 5G-enabled IoT, IEEE Access 9 (2021) 90075–90083.
[8] T. Veeramakali, R. Siva, B. Sivakumar, P. Senthil Mahesh, N. Krishnaraj, An intelligent internet of things-based secure healthcare framework using blockchain technology with an optimal deep learning model, J. Supercomput. 77 (9) (2021) 9576–9596.
[9] S. Sagar, A. Mahmood, Q.Z. Sheng, W.E. Zhang, Y. Zhang, J.K. Pabani, Understanding the trustworthiness management in the social internet of things: A survey, Comput. Netw. 251 (2024) 110611.
[10] C.A. Ardagna, R. Asal, E. Damiani, N.E. Ioini, M. Elahi, C. Pahl, From trustworthy data to trustworthy IoT: A data collection methodology based on blockchain, ACM Trans. Cyber- Phys. Syst. 5 (1) (2020) 1–26.
[11] L. Zhang, J. Liu, Y. Wei, D. An, X. Ning, Self-supervised learning-based multi-source spectral fusion for fruit quality evaluation: A case study in mango fruit ripeness prediction, Inf. Fusion 117 (2025) 102814.
[12] A.C. Sumathi, M. Akila, R. Pérez de Prado, M. Wozniak, P.B. Divakarachari, Dynamic bargain game theory in the internet of things for data trustworthiness, Sensors 21 (22) (2021) 7611.
[13] Z. Lv, Y. Han, A.K. Singh, G. Manogaran, H. Lv, Trustworthiness in industrial IoT systems based on artificial intelligence, IEEE Trans. Ind. Inform. 17 (2) (2020) 1496–1504.
[14] A. Sharma, E.S. Pilli, A.P. Mazumdar, P. Gera, Towards trustworthy internet of things: A survey on trust management applications and schemes, Comput. Commun. 160 (2020) 475–493.
[15] H.R. Hasan, K. Salah, I. Yaqoob, R. Jayaraman, S. Pesic, M. Omar, Trustworthy IoT data streaming using blockchain and IPFS, IEEE Access 10 (2022) 17707–17721.
[16] B. Soret, L.D. Nguyen, J. Seeger, A. Bröring, C.B. Issaid, S. Samarakoon, A. El Gabli, V. Kulkarni, M. Bennis, P. Popovski, Learning, computing, and trustworthiness in intelligent IoT environments: Performance-energy tradeoffs, IEEE Trans. Green Commun. Netw. 6 (1) (2021) 629–644.
[17] S. Prajapat, P. Kumar, D. Kumar, A.K. Das, M.S. Hossain, J.J. Rodrigues, Quantum secure authentication scheme for internet of medical things using blockchain, IEEE Internet Things J. (2024).
[18] D. Gautam, S. Prajapat, P. Kumar, A.K. Das, K. Cengiz, W. Susilo, Blockchain-assisted post-quantum privacy-preserving public auditing scheme to secure multimedia data in cloud storage, Cluster Comput. (2024) 1–14.
[19] S.-H. Sim, Y.-S. Jeong, Multi-blockchain-based IoT data processing techniques to ensure the integrity of IoT data in aIoT edge computing environments, Sensors 21 (10) (2021) 3515.
[20] K. Tulkinbekov, D.-H. Kim, Blockchain-enabled approach for big data processing in edge computing, IEEE Internet Things J. 9 (19) (2022) 18473–18486.
[21] L. Wang, Y. Wang, Supply chain financial service management system based on block chain IoT data sharing and edge computing, Alex. Eng. J. 61 (1) (2022) 147–158.
[22] H. Xue, D. Chen, N. Zhang, H.-N. Dai, K. Yu, Integration of blockchain and edge computing in internet of things: A survey, Future Gener. Comput. Syst. 144 (2023) 307–326.
[23] U. Sakthi, J. DafniRose, Blockchain-enabled smart agricultural knowledge discovery system using edge computing, Procedia Comput. Sci. 202 (2022) 73–82.
[24] Y. Zhu, C. Huang, Z. Hu, A. Al-Dhelaan, M. Al-Dhelaan, Blockchain-enabled access management system for edge computing, Electron. 10 (9) (2021) 1000.
[25] C. Luo, L. Xu, D. Li, W. Wu, Edge computing integrated with blockchain technologies, Complex. Approx.: Mem. Ker- I Ko (2020) 268–288.
[26] S. Prajapat, G. Thakur, P. Kumar, A.K. Das, M.S. Hossain, A blockchain-assisted privacy-preserving signature scheme using quantum teleportation for metaverse environment in web 3.0, Future Gener. Comput. Syst. 164 (2025) 107581.
[27] S. Prajapat, N. Kumar, A.K. Das, P. Kumar, R. Ali, Quantum-safe blockchain-assisted data encryption protocol for internet of things networks, Cluster Comput. 28 (1) (2025) 5.
[28] X. Ning, Z. Yu, L. Li, W. Li, P. Tiwari, DILF: Differentiable rendering-based multi-view image–language fusion for zero-shot 3D shape understanding, Inf. Fusion 102 (2024) 102033, http://dx.doi.org/10.1016/j.inffus.2023.102033, URL: https://www.sciencedirect.com/science/article/pii/S1566253523003494.
[29] M. Shafay, R.W. Ahmad, K. Salah, I. Yaqoob, R. Jayaraman, M. Omar, Blockchain for deep learning: review and open challenges, Cluster Comput. 26 (1) (2023) 197–221.
[30] J. Huang, X. Yu, D. An, X. Ning, J. Liu, P. Tiwari, Uniformity and deformation: A benchmark for multi-fish real-time tracking in the farming, Expert Syst. Appl. 264 (2025) 125653.
[31] Z. Shahbazi, Y.-C. Byun, Improving transactional data system based on an edge computing–blockchain–machine learning integrated framework, Process. 9 (1) (2021) 92.
[32] S. Rathore, J.H. Park, A blockchain-based deep learning approach for cyber security in next generation industrial cyber-physical systems, IEEE Trans. Ind. Inform. 17 (8) (2020) 5522–5532.

[33] G.N. Nguyen, N.H. Le Viet, M. Elhoseny, K. Shankar, B. Gupta, A.A. Abd El-Latif, Secure blockchain enabled cyber–physical systems in healthcare using deep belief network with ResNet model, J. Parallel Distrib. Comput. 153 (2021) 150–160.
[34] S. Wang, T. Wu, J. Li, L. Zhang, VC-DETR: A VGG-CNN-detection transformer model based blockchain secure iris detection, in: Proceedings of the 2024 Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence, 2024, pp. 880–884.
[35] A. Alzahrani, Novel approach for intrusion detection attacks on small drones using ConvLSTM model, IEEE Access (2024).
[36] J. Anin, M.J. Khan, O. Abdelsalam, M. Nabil, F. Hu, A. Alsharif, Efficient and privacy-preserving ConvLSTM-based detection of electricity theft cyber-attacks in smart grids, IEEE Access (2024).
[37] Mustaqeem, S. Kwon, CLSTM: Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network, Math. 8 (12) (2020) 2133.
[38] S.R.K. Somayaji, M. Alazab, M. Manoj, A. Bucchiarone, C.L. Chowdhary, T.R. Gadekallu, A framework for prediction and storage of battery life in IoT devices using DNN and blockchain, in: 2020 IEEE Globecom Workshops (GC Wkshps, IEEE, 2020, pp. 1–6.
[39] S. Sapkota, H. Huang, Y. Hu, F. Hussain, A deep neural network (DNN) based contract policy on hyperledger fabric for secure internet of things (IoTs), in: International Conference on Advanced Information Networking and Applications, Springer, 2024, pp. 313–325.
[40] J.M. Peterson, J.L. Leevy, T.M. Khoshgoftaar, A review and analysis of the bot-iot dataset, in: 2021 IEEE International Conference on Service-Oriented System Engineering, SOSE, IEEE, 2021, pp. 20–27.
[41] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, A. Anwar, TON_IoT telemetry dataset: A new generation dataset of IoT and iIoT for data-driven intrusion detection systems, Ieee Access 8 (2020) 165130–165150.
[42] G. Chen, J. Wu, W. Yang, A.K. Bashir, G. Li, M. Hammoudeh, Leveraging graph convolutional-LSTM for energy-efficient caching in blockchain-based green IoT, IEEE Trans. Green Commun. Netw. 5 (3) (2021) 1154–1164.
[43] S.K. Singh, M. Kumar, S. Tanwar, J.H. Park, GRU-based digital twin framework for data allocation and storage in IoT-enabled smart home networks, Future Gener. Comput. Syst. 153 (2024) 391–402.
[44] A. Bourechak, O. Zedadra, M.N. Kouahla, A. Guerrieri, H. Seridi, G. Fortino, At the confluence of artificial intelligence and edge computing in iot-based applications: A review and new perspectives, Sensors 23 (3) (2023) 1639.
[45] S. Ma, J. Nie, J. Kang, L. Lyu, R.W. Liu, R. Zhao, Z. Liu, D. Niyato, Privacy-preserving anomaly detection in cloud manufacturing via federated transformer, IEEE Trans. Ind. Inform. 18 (12) (2022) 8977–8987.
[46] E.C. Nkoro, C.I. Nwakanma, J.-M. Lee, D.-S. Kim, Bit-by-bit: A quantization-aware training framework with XAI for robust metaverse cybersecurity, in: 2024 International Conference on Artificial Intelligence in Information and Communication, ICAIIC, IEEE, 2024, pp. 832–837.
[47] C.S. Prasanna, M.Z.U. Rahman, M.D. Bayleyegn, Brain epileptic seizure detection using joint CNN and exhaustive feature selection with RNN-BLSTM classifier, IEEE Access 11 (2023) 97990–98004.