Edge Intelligence for Smart Metro Systems: Architecture and Enabling Technologies

Xing Liu, Minjie Zhang, Chengming Zou*
Wuhan University of Technology
Wuhan, China
liu.xing@whut.edu.cn

Jianfeng Yang*
Electronic Information School
Wuhan University
Wuhan, China
yjf@whu.edu.cn

Xin Yan
Wuhan University of Technology
Wuhan, China
yanxin@whut.edu.cn

ABSTRACT

Safety and operational efficiency (SOE) are of great significance for the development of a smart metro system (SMS). To improve the SOE of SMS, Internet-of-Things technology is applied first to collect metro environmental data (MED), and then these data are analyzed by intelligent algorithms to achieve safety risk prediction, defect detection or operational efficiency improvement. As MEDs are generated quickly in the practical engineering field and most algorithms to process these data also have high computational complexity, the SMS must have sufficient computing power to process these MEDs in time. However, modern train computing resources are insufficient to meet this requirement. To address this challenge, edge intelligence (EI) technology, which enables trains to offload computationally-intensive tasks to nearby EI systems, is proposed. This paper aims to develop an energy-efficient and high-performance EI system to improve the SOE of SMSs. An EI architecture that considers the three driving forces, data, algorithms and computing power, is proposed first, and then the enabling techniques of this EI architecture, including intelligent algorithms, domain-specific architecture (DSA)-based hardware acceleration, end-edge-cloud collaborative computing, hardware platform management, and security issues, are investigated. This EI system can enable SMS to process the large-scale MEDs with not only high accuracy but also low latency and low energy cost. As a study case, a real-world EI system is built to run three kinds of SMS applications to assess the safety risks of SMSs. The evaluation results demonstrate the effectiveness of the proposed schemes in this paper.

Keywords

metro, edge intelligence, architecture, real time, energy efficiency

1. INTRODUCTION

A smart metro system (SMS) that not only detects abnormal events or predicts safety risks but also maximizes the operational efficiency (OE) to minimize both passenger waiting time and train operating cost is of great significance to metro system development.

To develop such an SMS, Internet-of-Things (IoT) technology can be applied first to collect metro environmental data (MED), and then these MEDs can be analyzed by intelligent algorithms to achieve functionalities such as safety risk prediction and OE improvement.

In recent years, many studies have applied the above approaches to improve the safety and OE (SOE) of metro systems [1, 2, 3, 4]. Although many of them are effective in theory, they cannot function well in real-world scenarios. This is because most of these studies focus on optimizing algorithm accuracy but neglect the significance of algorithm data processing speed. However, in the practical engineering field, large numbers of IoT devices work continuously with high sampling frequency. Consequently, a large quantity of MED is generated within a short time. These MED need to be processed with high speed. Otherwise, the data that cannot be processed immediately will gradually accumulate and lose timeliness. Therefore, the MED processing speed is significant for SMSs, and it should be faster than the MED generating speed. This requires the SMS not only to have proper algorithms to process the MED accurately but also to have sufficient computing power to enable the algorithms to process the MED quickly. In other words, the design of an SMS should consider three indispensable driving forces: data, algorithms and computing power.

Although modern trains have richer computing power and storage resources, they are still not competent for processing large-scale MED with high speed when running computationally-intensive algorithms (CIAs). Therefore, it is inappropriate for the SMS trains to directly process the MED. Traditional methods resort to resource-rich cloud datacenters to perform these computationally-intensive tasks, yet they may cause intolerable end-to-end latency and heavy energy consumption due to network congestion between trains and remote cloud data centers.

In recent years, edge intelligence (EI) technology has been envisioned as a promising technology to address the above challenges [5]. EI enables resource-constrained SMS devices to offload partial data and computationally-intensive tasks to nearby edge servers so that the risk of network congestion can be diminished and large-scale MED can be processed quickly within deadlines. In Figure 1, an EI-empowered SMS is depicted. Large numbers of IoT devices are equipped

* Authors to whom all the correspondences should be addressed. This work was supported in part by the Natural Science Foundation of Xinjiang Province of China under Grant 2020D01A130, in part by the National Natural Science Foundation of China under Grant 61702387 and 61771354.

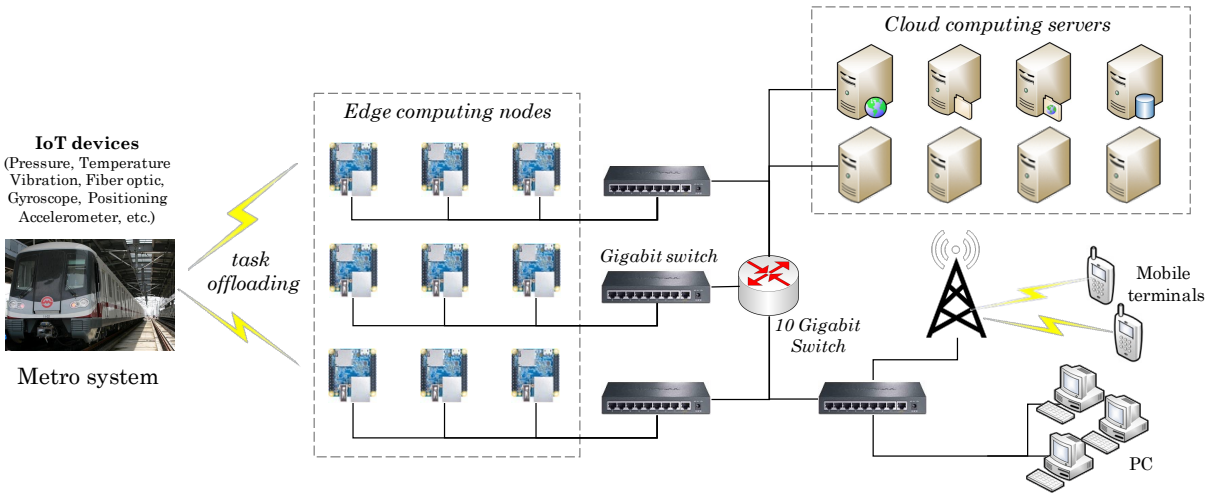

Figure 1: An overview of EI-empowered smart metro systems.

on trains, under tracks or on tunnel walls to collect MED. Most of these MED are offloaded to the EI system and then analyzed by EI system intelligent algorithms. Finally, the analysis results are returned to the SMS to enable it to make intelligent decisions.

As discussed above, the EI system for an SMS should have sufficient computing power to enable the large-scale MED to be processed quickly. To meet this requirement, most traditional methods simply increase the number of computing units. This method is effective. However, it can cause high device costs and considerable energy costs and is not appropriate for application in many real-world engineering applications that require the system to be economical.

To strengthen the computing power while maintaining the high energy efficiency of EI systems, the following strategies are implemented in our work. (1) A domain-specific architecture (DSA) technique [6] is used to design hardware architectures tailored to CIAs in SMSs, which allows significant performance and efficiency gains. (2) An end-edge-cloud collaborative computing (ECCC) mechanism is implemented to integrate the computing resources of clouds, edges and end devices. (3) Other mechanisms, such as virtualization, lightweight containers and heterogeneous scheduling, are implemented to improve the computing system efficiency.

The objective of this paper is to develop an energy-efficient and high-performance EI system to improve the SOE of SMSs. This EI system can enable SMS to process the large-scale MEDs with not only high accuracy but also low latency and low energy cost.

The contributions of this paper are various. First, we design an EI architecture for SMS that considers all three indispensable driving forces: data, algorithms and computing power. Second, we investigate the key enabling technologies applied in this EI architecture, including intelligent algorithms, DSA-based computing acceleration, ECCC, hardware management strategies and security issues. Third, a real-world EI system is implemented to evaluate the performance of the proposed schemes.

The rest of this article is organized as follows: In Section 2, the EI architecture for SMSs is proposed. In Section 3, we discuss the key enabling technologies applied in this EI architecture. In Section 4, a real-world EI system is built to evaluate the performance of the proposed mechanisms. Finally, in Section 5, the conclusion is given.

2. EDGE INTELLIGENCE ARCHITECTURE FOR SMS

The EI architecture for SMSs is depicted in Figure 2. This section discusses the design concepts of this EI architecture.

First, in the application layer, a set of SMS applications are deployed to improve the SOE of the SMS. Then, in the algorithm layer, the algorithms that should be used by these SMS applications are implemented, including neural networks (NNs), Bayesian networks (BNs), support vector machines (SVMs), and deep learning (DL).

As many algorithms in EI systems have high computational complexity and the data to be processed by these algorithms are also on a large scale, the EI system is required to have sufficient computing power. Therefore, in the hardware layer, DSA technology is applied by equipping the EI system with heterogeneous DSA-based computing units such as FPGA accelerators, DL processors and ASICs. This method can accelerate the data processing speed of the CIAs while maintaining low EI system energy costs.

Once the heterogeneous computing architecture is applied in the hardware layer, it creates several new challenges, including how to schedule the tasks among different computing units efficiently and how to deploy the tasks in the heterogeneous system more easily. Therefore, between the algorithm layer and hardware layer, a platform management layer is designed to address these challenges.

In addition, the ECCC mechanism is applied to enable device-edge joint computing and edge-cloud collaborative computing. Furthermore, the security defense framework for edge computing is also proposed in this architecture.

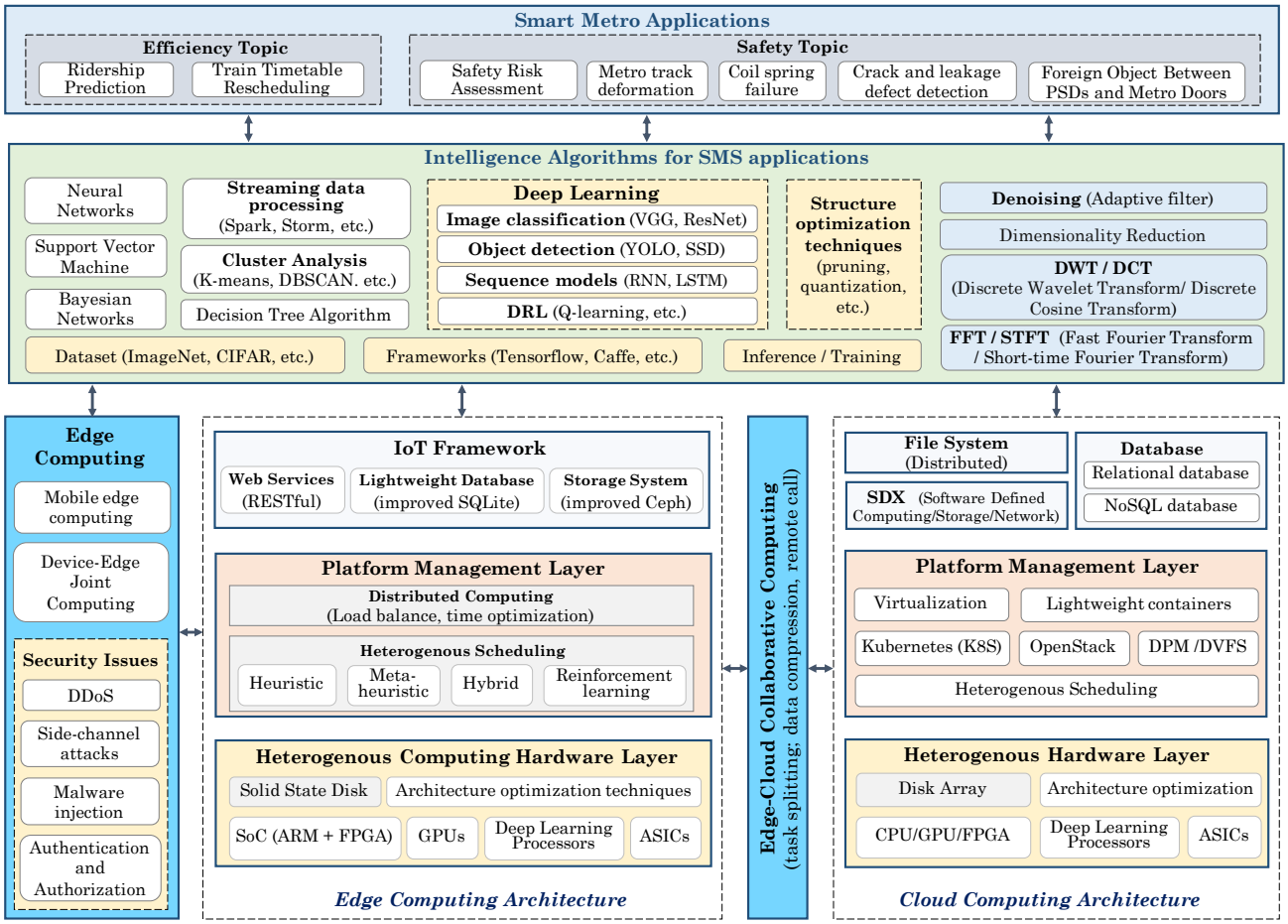

Figure 2: Edge intelligence architecture for smart metro systems.

3. ENABLING TECHNOLOGIES OF EDGE INTELLIGENCE ARCHITECTURE

This section presents the key enabling technologies of the proposed EI architecture for the SMS.

3.1 SMS Applications and Algorithms

Currently, many SMS applications have been developed to improve SMS SOE. This section introduces several representative applications and their algorithms.

3.1.1 Ridership Prediction using Neural Networks

Accurate metro ridership prediction (MRP) can not only guide passengers to select departure times reasonably but also guide metro authorities to allocate limited resources effectively.

Since many MRP works have less consideration of spatiotemporal features, Ma et al. [1] proposed a parallel architecture comprising a convolutional neural network (CNN) and a bidirectional long short-term memory network (BLSTM) to extract spatial and temporal features of MRP, respectively. The experimental results showed that the proposed model outperforms traditional prediction approaches and is suitable for MRP in large-scale metro networks.

3.1.2 Timetable Rescheduling using Neural Networks

Train timetable rescheduling (TTR) needs to be performed when incidents or perturbations occur, and an efficient TTR can minimize both passenger waiting time and train operating costs.

Ying et al. [2] studied a TTR with rolling stock circulation under stochastic demand. Aiming at the problem that the scheduling process complexity increases exponentially when states, decisions and uncertainties are involved, they proposed a novel actor-critic deep reinforcement learning approach to simplify the search process for potential optimal solutions. Experimental results showed that the proposed mechanism outperforms a range of metaheuristic approaches in terms of computing time, efficiency and robustness.

3.1.3 Defect Detection using Neural Networks

Traditionally, detecting cracks and leakage defects (CLDs) in metro shield tunnels is performed by handcrafted features. However, this method has low efficiency in distinguishing defects from some interferences, such as segmental joints, bolt holes, cables and manual marks. To achieve rapid and accurate defect recognition, Huang et al. [3] proposed a new

image recognition algorithm for CLD semantic segmentation using hierarchies of features extracted by a fully convolutional network (FCN). Experimental results illustrated that the proposed method has great superiority in recognition results, inference time and error rates compared with the frequently used traditional methods.

3.1.4 Safety Risk Assessment using Bayesian Networks and SVM

BN and SVM are effective methods for facilitating knowledgeable reasoning in uncertain environments. Thus, Wang et al. [4] used a BN to assess the safety risks in metro construction. Considering that BN analysis has difficulty obtaining exact probability values in construction engineering fields due to the lack of sufficient data, they combined BN with the fuzzy comprehensive evaluation method (FCEM). Since FCEM can solve engineering problems under uncertainty by utilizing intervals or fuzzy numbers, the combination of these two approaches can systematically assess the inherent safety risks associated with metro construction.

3.2 Computing Acceleration in EI Systems

Many algorithms used in SMS have high computational complexity, and the data processed by these algorithms are also on a large scale. Therefore, computing acceleration is critical to SMSs. In this section, we use computationally-intensive deep NN (DNN) algorithms as representative CIAs in SMSs to discuss how to use the software and hardware co-optimization mechanism to accelerate the DNNs in EI systems.

3.2.1 Software Acceleration: DNN Structure Optimization

DNN execution can be accelerated from the software perspective by performing DNN structure optimization [7].

Network pruning calculates the importance of neurons in terms of pruning criteria and prunes the least important neurons to decrease the computational complexity and parameter size. Network quantization decreases the computing complexity and parameter size by performing a high compression ratio or substituting float-point arithmetic with low-precision fixed-point arithmetic. The teacher-student network trains a compact student network with the help of a large teacher network and transfers dark knowledge from the teacher network to the student network to enable the student network to achieve higher accuracy. Tensor decomposition decomposes the 4D convolutional tensor into a set of low-dimensional tensors to simplify the computational complexity of convolutional operations.

3.2.2 Hardware Acceleration: DL Processors

Traditionally, DNN algorithms are executed on CPUs and GPUs. However, CPUs are inefficient for performing DNN computations because they cannot take advantage of the intrinsically parallel computing characteristics of DNNs. GPUs have improved throughput over CPUs, yet their naturally high energy consumption restricts their large-scale deployment in EI systems. To accelerate DNNs while maintaining high energy efficiency, DSA technologies such as DL processors or FPGA-based accelerators are popularly studied.

DL processors design the hardware architecture specifically in terms of the computing features and memory operating characteristics of DL algorithms, thereby offering better computing performance and higher energy efficiency compared with the general-purpose CPUs and GPUs. The DianNao family series and Google tensor processing units (TPUs) are well-known DL processors.

3.2.3 Hardware Acceleration: FPGA-based Deep Learning Accelerators

FPGA provides thousands of programmable logic elements and programmable interconnects, which enables it to build custom-tailored accelerators for DNNs.

In recent years, FPGA-based DNN accelerators have drawn increasing attention, and many DNN architecture design technologies, such as systolic arrays, parallelism, pipelining, loop reordering, loop unrolling, tilling, batching and ping-pong double buffers, have been studied [8].

Extensive studies show that FPGA-based DNN accelerators can achieve lower latency and higher energy efficiency than general-purpose CPUs [8]. Moreover, they have high hardware flexibility and can flexibly adapt to the new emerging DL algorithms.

3.3 End-Edge-Cloud Collaborative Computing

ECCC can integrate the advantages of edge computing (EC)'s low latency and cloud computing (CC)'s high performance. This section presents methods for using ECCC to optimize CIA computing latency in SMSs, and the DL algorithm is also representative of CIAs for this discussion.

3.3.1 Task Offloading Strategies for Mobile Edge Computing

Many SMS applications are composed of dependent tasks, and most existing works use heuristic or approximate algorithms to solve the task offloading problem for these applications. However, heuristic algorithms depend heavily on accurate mathematical models and are inappropriate for increasingly dynamic and complex SMS scenarios.

To address the above challenges, many recent works have proposed a deep reinforcement learning (DRL)-based offloading strategy. DRL is appropriate for addressing the offloading problem in dynamic mobile edge computing (MEC) scenarios since it can learn the optimal offloading policy by treating the complex MEC system as a black box and interacting directly with the MEC environment in a trial-and-error manner without building accurate models of the environment.

Chen et al. [9] considered MEC for a representative mobile user in an ultradense sliced radio access network. They model the optimal computation offloading policy as a Markov decision process first. Then, to address the challenge of high dimensionality in state space, they proposed a double deep Q-network (DQN)-based algorithm to learn the optimal policy without acquiring a priori knowledge of network dynamics. Furthermore, they combined DQN with Q-function decomposition to solve stochastic computational offloading. The results showed a significant improvement in computa-

tional offloading performance compared with the baseline policies.

3.3.2 Device-Edge Joint Computing

Running computationally-intensive tasks on resource-limited mobile devices is infeasible, while offloading all these tasks to edge servers may also suffer from time-varying wireless fading channel problems. To address these challenges, the device-edge joint computing mechanism can be used.

Li et al. [10] proposed an edge artificial intelligence (AI) framework that leverages EC for DNN collaborative inference through device-edge synergy. By observing the phenomenon that a DNN network layer with higher latency may not output a larger data size, the DNN partitioning technique, which adaptively partitions computation between devices and edges and offloads the computationally-intensive part to the MEC server, is applied to achieve real-time DNN inference. Experimental results show that DNN inference tasks can be completed with low latency while maintaining high accuracy.

3.3.3 Edge-Cloud Collaborative Computing

Edge-cloud collaborative computing (ECC) can extend the processing capabilities of edge servers by offloading partial computation tasks to remote cloud servers. The implementation of ECC in DL-based applications usually involves the task splitting operation, which divides the NN into an edge partition and a cloud partition, and the data compression operation, which compresses the data at the edge before being sent to the cloud.

Currently, many studies have applied ECC to optimize the computation of DL. Gu et al. [11] proposed an ECC architecture that resorts to the clouds for object tracking performance enhancement. By properly offloading partial computational tasks to the cloud and periodically checking the tracking status of edge devices through convolutional Siamese networks, the tracking errors can be rectified quickly and accurately. Moreover, the energy cost of edge devices can be saved.

3.4 Hardware Platform Management

This section presents the hardware management strategies that can be used to improve the utilization efficiency or computing scalability of the hardware resources in EI systems.

3.4.1 Energy-efficient Time-aware Heterogeneous Scheduling

Since the heterogeneous computing architecture has been applied in the EI system hardware layer, an energy-efficient and time-aware heterogeneous scheduling mechanism needs to be studied in the hardware management layer.

Heuristics and metaheuristics are the two basic kinds of heterogeneous scheduling algorithms. Some studies attempted to combine two or more heuristic or metaheuristic algorithms, named hybrid scheduling, to strengthen the scheduling performance in convergence speed, makespan, energy efficiency and resource cost.

In recent years, RL has also been popularly applied in heterogeneous scheduling to overcome the limitations in traditional task scheduling approaches [12]. RL is appropriate for application in this scheduling problem because it does not need to learn from the labeled training set and can obtain the task scheduling model by offline training in terms of the environmental feedback information.

3.4.2 Computing Resource Virtualization

Computing resource virtualization is significant for improving EI system hardware utilization efficiency, increasing computing scalability, and strengthening security.

There has been much CPU virtualization work in the past. However, GPU virtualization is a relatively new study field due to the intellectual property protection of GPU drivers. FPGA virtualization is different from that of CPUs and GPUs in that the FPGA applications are implemented in hardware circuits rather than instructions. This makes traditional software-based virtualization techniques infeasible for direct application to FPGAs. In article [13], the differences between FPGA virtualization techniques and equivalent software virtualization techniques are compared.

3.4.3 Parallel and Distributed Computing

Parallel and distributed computing can maximize the computing performance of EI by connecting different computing nodes in a cost-effective, transparent and reliable manner.

In the EI cloud computing system, Kubernetes, also known as K8s, is deployed to realize distributed computing. K8s is chosen because it can automate the deployment, scaling and management of containerized applications. It can group the containers into logical units to achieve easy management and distributed computing. It also has extensive support for CPUs, GPUs and FPGAs.

3.5 Security Threats for Edge Computing

As most developers tend to focus on performance optimization rather than security threats when designing EC architectures, the defense against EI system security attacks is significant.

3.5.1 Security Challenges in Edge Computing

Compared with the CC system, the EC system introduces more security threats for the following reasons:

(1) Weakened computing power: EC systems have weaker computing power and are mostly equipped with more fragile defense systems than CC systems, which makes EC systems more vulnerable to security attacks.

(2) Nonmigratability of security frameworks: The operating systems and communication protocols of EC devices are mostly heterogeneous and lack standardized regulation, which not only makes it hard to migrate the security frameworks from fledged general-purpose computers (GPCs) to ECs but also from one EC device to another.

(3) Coarse-grained access control: the access control models designed for GPCs and CCs cannot be satisfied for ECs since EC systems have more complicated enabled applications. Fine-grained access control permissions specific to ECs are essential to defend against security attacks.

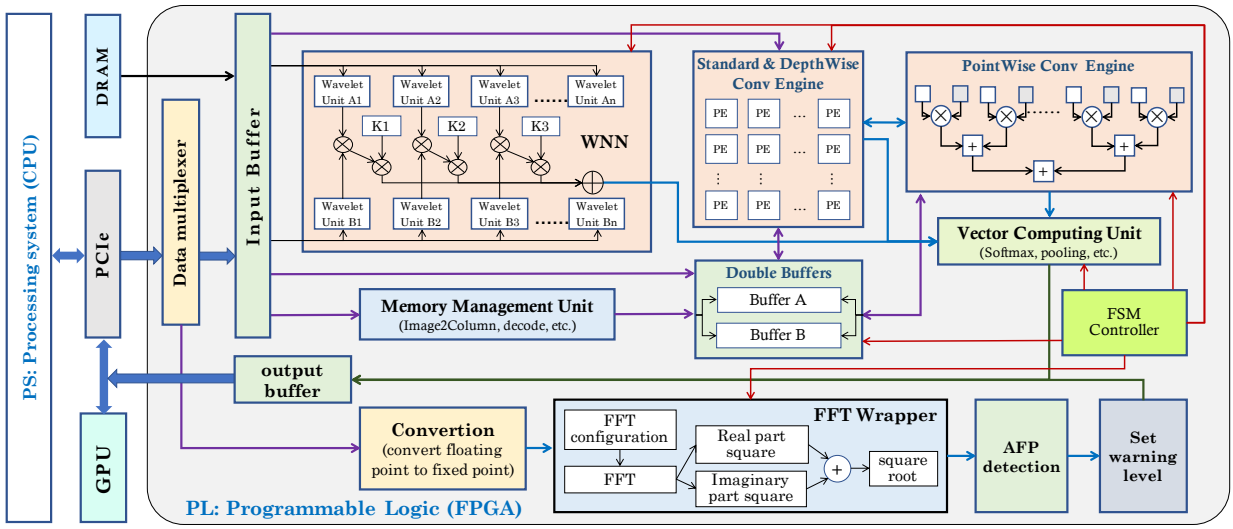

Figure 3: Block diagram of FPGA-based accelerators for accelerating CIA in SMS.

3.5.2 Types of Security Attacks in Edge Computing

The major security attacks applied to EC applications can be classified into four categories: distributed denial of service attacks (DDoS) attacks, side-channel attacks, malware injection attacks, and authentication and authorization attacks [14]. These four types of attacks account for 82% of the security attacks in the EC system [14]. In article [14], a detailed review of the attack specifications and defense solutions of these four types of attacks is given.

In addition to the above attacks, the bad data injection attack is also not trivial when the EC system uses AI and machine learning algorithms. In these EC systems, the training models are mostly trained by the collected data; thus, an attacker can launch the attack by submitting fake data to corrupt the training models. To defend against this attack, article [15] introduced a federated learning mechanism, which prevents the data from leaving its local client and prevents the model parameters from leaving the private network. In this way, security data analysis might be realized.

3.5.3 Root Causes of Security Threats

To find solutions to defend against security threats, it is essential to understand the main root causes of security threats in the EC system, which can be summarized as follows [14]:

(1) Protocol design flaws: Many EC protocols have design flaws since designers mostly focus on functionalities and user experience and do not sufficiently consider security issues.

(2) Code-level flaws and implementation-level vulnerabilities: These flaws may include logic flaws in a practical realization, programming bugs that can cause memory crashes. The attackers can exploit these vulnerabilities to perform attacks such as bypassing authentication or malware injection.

(3) Data correlations: The correlations between public data and private data can enable attackers to use public data to infer private information or even tamper with it.

(4) Coarse-grained access control: EC systems have more complex permission scenarios and require fine-grained access control. However, many EC systems only implement coarse-grained access control or do not implement any access control mechanisms, which lowers the security attack threshold.

4. PERFORMANCE EVALUATION

In the experiment, we implement an energy-efficient and high-performance EI system to run three kinds of SMS applications to predict SMS safety risks.

4.1 Experimental setup

4.1.1 SMS Application Layer

In the application layer, three kinds of SMS applications are developed: foreign object detection in metro tunnels, safety risk prediction for urban constructions around metro stations, and metro track deformation detection. These applications can predict SMS safety risks, thereby avoiding safety accidents.

The MEDs used by these applications are collected by a set of optical fiber sensors (OFSs) laid under metro tracks and by many other IoT devices, such as temperature sensors, gyroscopes, accelerometers, positioning and cameras that are equipped on trains.

4.1.2 Intelligent Algorithm Layer

In the algorithm layer, three kinds of algorithms are developed for the above three SMS applications: SSD-MobileNet, fast Fourier transform (FFT) and wavelet NN (WNN).

SSD-MobileNet is used for object detection and is improved based on the single short detection (SSD) algorithm by substituting the standard convolution of SSD with MobileNet.

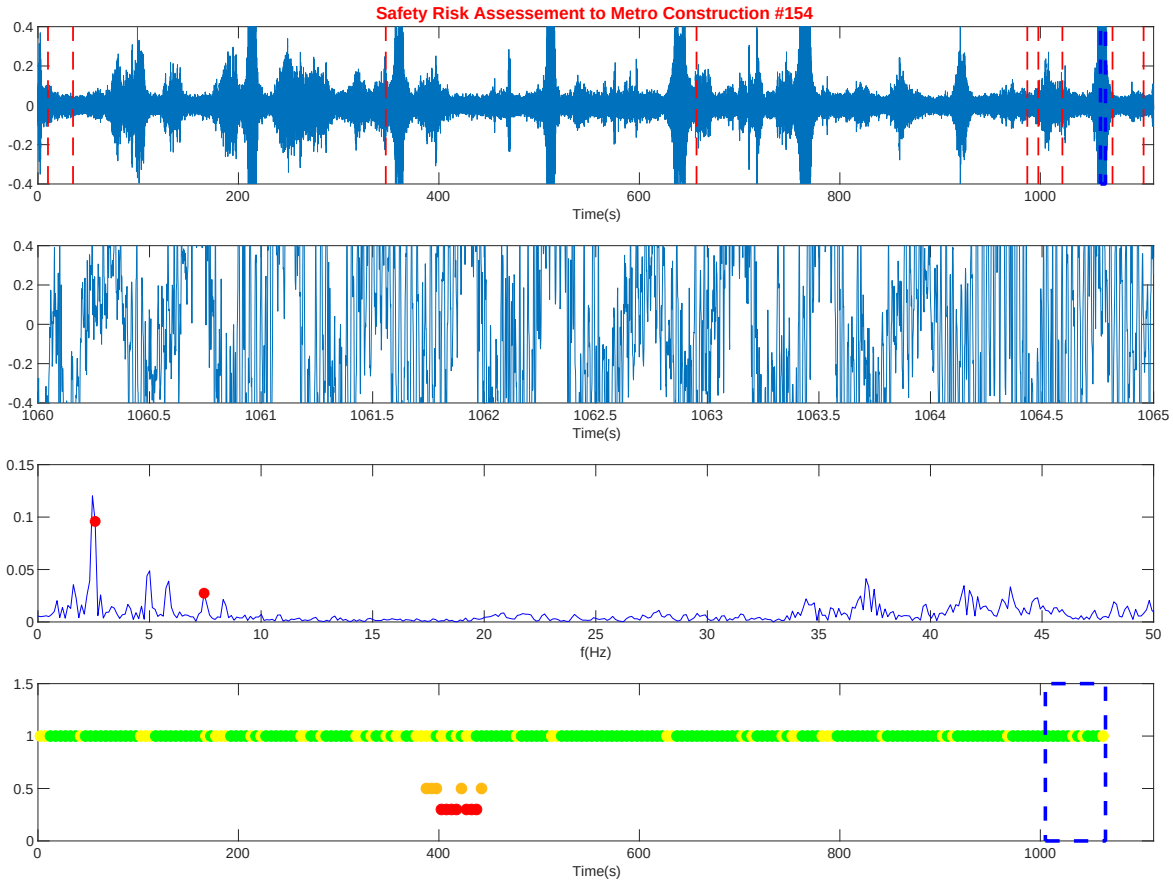

Figure 4: MED analysis for safety risk assessment of the constructions adjacent to metro stations.

The FFT is used to assess the construction safety risks near the metro stations. It analyzes the data collected from OFSs and IoT devices and searches for abnormal frequency points (AFPs) from two frequency domains: 2.5 Hz ~ 23.6 Hz and 25.6 Hz ~ 35 Hz. The AFPs refer to the frequency points of which the amplitude is 3 times greater than the average amplitude of the frequency range located surrounding 1 Hz of this point, and the above two frequency domains are selected because the vibration frequency of drilling rigs is usually within these domains.

The WNN is used for track deformation detection and is improved based the NN algorithm by substituting the nonlinear transformation functions of hidden nodes in NN with the Morlet wavelet functions. Once the tracks are deformed, the transmission characteristics of light inside the optical fibers change. By collecting the optical signal data with OFSs and analyzing these data by FPGA-based WNN accelerators, the track deformation can be assessed.

4.1.3 Heterogeneous Hardware Layer

In the hardware layer, the DSA technique is applied by designing several FPGA-based accelerators dedicated to the SSD-MobileNet, FFT and WNN algorithms. These accelerators can be integrated inside one FPGA card and can work in parallel to process the input data of different SMS applications by using the data multiplexer technique. In Figure 3, block diagrams of these accelerators are depicted.

To improve the computing performance and energy efficiency of these accelerators, a set of optimization techniques are applied to design the architectures of these accelerators, such as systolic arrays, double buffers, parallelism and pipelines.

For the MobileNet accelerator, two independent convolutional computing engines dedicated to the depthwise convolutions and pointwise convolutions are designed respectively to increase the throughput of MobileNet.

When designing the WNN accelerator on FPGA, it meets two challenges: difficulty using an electronic circuit to implement gradient descent and becoming trapped in a local minimum. The first problem can be solved by using the Taylor series (TS) to approximate the values of nonlinear activation functions, while the second problem can be alleviated by combining WNN with particle swarm optimization (PSO), named WNN-PSO, and using PSO to train the parameters of WNN.

Considering that different computing architectures have different advantages, FPGA accelerators are further combined with CPUs and GPUs to build a heterogeneous CPU/GPU/FPGA computing platform so that EI system hardware can flexibly adapt to different SMS contexts.

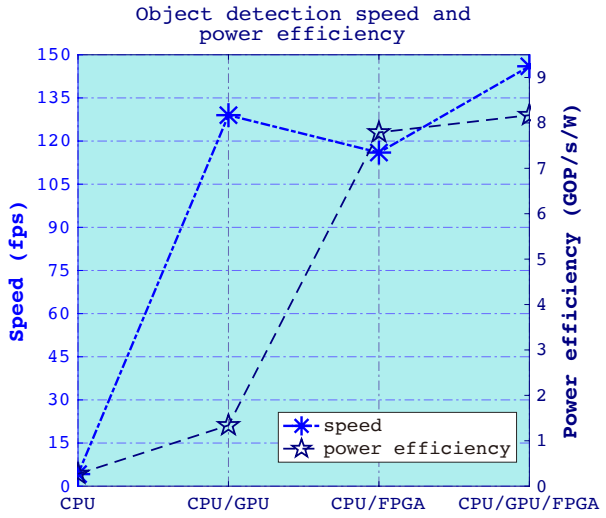

Figure 5: Speed and power efficiency of MobileNet in an EI system.

4.1.4 Hardware Platform Management Layer

In the hardware platform management layer, we applied our previous heterogeneous task scheduling work in [12] to assign each SMS task to the most appropriate core on the heterogeneous CPU/GPU/FPGA system. In addition, we implemented Kubernetes (k8s) and Docker technologies to achieve distributed computing among different heterogeneous computing nodes.

4.2 Experimental Results

As a key challenge in EI systems is to provide sufficient computing power to process large-scale MEDs with high speed and low energy cost, the performance evaluation of EI systems in this experiment focuses on two indicators: computing power and energy efficiency; computing power can also be evaluated by the data processing speed.

4.2.1 Performance Evaluation of the EI System for Construction Safety Risk Assessment

The safety risk assessment of the constructions adjacent to the metro stations is performed by FFT. In Figure 4, the top three subfigures depict the time domain and frequency domain of MED, and the bottom subfigure shows the AFPs that have been detected by FFT. The AFPs are marked in red, and the larger the number of AFPs is, the higher the metro station safety risk level.

In the experiment, we run FFT to analyze the OFS data on the ARM Cortex-A7 CPU and Zynq7020 FPGA accelerator. The results show that the ARM CPU requires 112µs to process 5,000 sets of data by FFT, while the FPGA accelerator only requires 21µs when the parallelism of processing elements is set to 2. This result demonstrates the effectiveness of using an FFT accelerator in an EI system.

4.2.2 Performance Evaluation of the EI System for Foreign Object Detection

Table 1: Performance evaluation of the WNN accelerator in the EI system

Titles	Platforms	Values
Prediction RMS errors	software	0.00092
	FPGA hardware	0.003312
Data processing speed per node (MB/s)	ARM Cortex-A7	28.6
	Zynq7020 FPGA	186.7
Power efficiency (GOP/s/W)	ARM Cortex-A7	0.36
	Zynq7020 FPGA	7.2

Foreign object detection is realized in EI by SSD-MobileNet. In the experiment, we use the COCO dataset and a Xeon E5 CPU, an NVIDIA Tesla P100 GPU and Intel Stratix 10 FPGA accelerators to conduct the following experiments: (1) Run SSD-MobileNet on CPUs. (2) Run SSD-MobileNet on CPU/GPU. (3) Run SSD-MobileNet on CPU/FPGA. (4) Run SSD-MobileNet on CPU/GPU/FPGA, that is, run MobileNet on the FPGA accelerator and run the other SSD tasks on the GPU.

The speed and energy cost of different experimental cases are depicted in Figure 5. The results show that FPGA-based accelerators achieve better computing performance than the CPU and much higher power efficiency than the GPU. The results also show that the CPU/GPU/FPGA computing platform achieves better computing performance and energy efficiency than the CPU/GPU or CPU/FPGA.

4.2.3 Performance Evaluation of the EI system for Track Deformation Prediction

Track deformation prediction is realized in the EI by the WNN-PSO algorithm. In the experiment, we run PSO on the GPU to train the WNN and then run the trained WNN model on the ARM Cortex-A7 CPU and Zynq7020 FPGA accelerators to process the OFS data. The evaluation result is shown in Table 1.

The results show that the EI system equipped with an FPGA-based WNN accelerator can achieve higher power efficiency and computing performance than ARM CPUs. Regarding the prediction root-mean-square (RMS) errors, the FPGA is slightly lower than the software implementation because it has used TS to approximate the values of activation functions. However, the prediction performance remains high.

5. CONCLUSION

This article proposed an energy-efficient and high-performance EI system to improve SMS SOE. This EI system considers all three indispensable driving forces of SMS, data, algorithms and computing power, and applies a set of emerging techniques, such as AI algorithms, DSA-based hardware acceleration, ECCC, heterogeneous scheduling and distributed computing. Real-world experiments demonstrate that the proposed EI system can achieve significant performance and efficiency gains and can effectively improve SMS SOE.

6. ACKNOWLEDGMENT

This work is supported in part by the Natural Science Foundation of Xinjiang Province of China (No. 2020D01A130); the National Natural Science Foundation of China under (No. 61702387, 61771354); National Innovation and Entrepreneurship Training Program for College Students (No. 202110497040). Chengming Zou and Jianfeng Yang are the corresponding authors.

7. BIOGRAPHIES

Xing Liu received the Ph.D. degree from the LIMOS Laboratory, University Blaise Pascal, Clermont- Ferrand, France in 2014. He is currently an Associate Professor with the School of Computer Science and Technology, Wuhan University of Technology, Wuhan. His research interests include the Internet of Things, artificial intelligence, and embedded systems.

Minjie Zhang received the bachelor degree from School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China in 2018. He is currently a graduate student with School of Computer Science and Technology, Wuhan University of Technology. His research interests include the Internet of Things, artificial intelligence, and embedded systems.

Chengming Zou received the Ph.D. degree from the Wuhan University of Technology, Wuhan, China, in 2003. He is currently a Professor with the School of Computer Science and Technology, Wuhan University of Technology. His research interests include the Internet of Things, artificial intelligence, and embedded systems.

Jianfeng Yang received the Ph.D. degree from the School of Electronic Information, Wuhan University, Wuhan, China, in 2009. He is currently an Associate Professor with Wuhan University, where he leads the Internet and IT Laboratory. His research interests include networking, edge computing, and high-reliability real-time wireless communication.

Xin Yan received the Ph.D. degree from the Wuhan University of Technology, Wuhan, China, in 2006. He is currently an Associate Professor with the School of Computer Science and Technology, Wuhan University of Technology. His research interests include mobile computing, and wireless networking and smart grid communications.

8. REFERENCES

[1] X. Ma, J. Zhang, B. Du, C. Ding, and L. Sun, "Parallel architecture of convolutional bi-directional lstm neural networks for network-wide metro ridership prediction," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 6, pp. 2278–2288, 2018.

[2] C.-s. Ying, A. H. Chow, and K.-S. Chin, "An actor-critic deep reinforcement learning approach for metro train scheduling with rolling stock circulation under stochastic demand," Transportation Research Part B: Methodological, vol. 140, pp. 210–235, 2020.

[3] H.-w. Huang, Q.-t. Li, and D.-m. Zhang, "Deep learning based image recognition for crack and leakage defects of metro shield tunnel," Tunnelling and underground space technology, vol. 77, pp. 166–176, 2018.

[4] Z. Wang and C. Chen, "Fuzzy comprehensive bayesian network-based safety risk assessment for metro construction projects," Tunnelling and Underground Space Technology, vol. 70, pp. 330–342, 2017.

[5] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," Proceedings of the IEEE, vol. 107, no. 8, pp. 1738–1762, 2019.

[6] J. L. Hennessy and D. A. Patterson, "A new golden age for computer architecture," Communications of the ACM, vol. 62, no. 2, pp. 48–60, 2019.

[7] L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," Proceedings of the IEEE, vol. 108, no. 4, pp. 485–532, 2020.

[8] K. Guo, S. Zeng, J. Yu, Y. Wang, and H. Yang, "[dl] a survey of fpga-based neural network inference accelerators," ACM Transactions on Reconfigurable Technology and Systems (TRETS), vol. 12, no. 1, pp. 1–26, 2019.

[9] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4005–4018, 2018.

[10] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge ai: On-demand accelerating deep neural network inference via edge computing," IEEE Transactions on Wireless Communications, vol. 19, no. 1, pp. 447–457, 2019.

[11] H. Gu, Z. Ge, E. Cao, M. Chen, and S. Hu, "A collaborative and sustainable edge-cloud architecture for object tracking with convolutional siamese networks," IEEE Transactions on Sustainable Computing, vol. 6, no. 1, pp. 144–154, 2021.

[12] X. Liu, J. Yang, C. Zou, Q. Chen, and C. Cai, "Collaborative edge computing with fpga-based cnn accelerators for energy-efficient and time-aware face tracking system," IEEE Transactions on Computational Social Systems, vol. PP, no. 99, pp. 1–15, 2021.

[13] A. Vaishnav, K. D. Pham, and D. Koch, "A survey on fpga virtualization," in 2018 28th International Conference on Field Programmable Logic and Applications (FPL), pp. 131–1317, IEEE, 2018.

[14] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: State of the art and challenges," Proceedings of the IEEE, vol. 107, no. 8, pp. 1608–1631, 2019.

[15] M. Mukherjee, R. Matam, C. X. Mavromoustakis, H. Jiang, G. Mastorakis, and M. Guo, "Intelligent edge computing: Security and privacy challenges," IEEE Communications Magazine, vol. 58, no. 9, pp. 26–31, 2020.