FIELD-PROGRAMMABLE GATE ARRAY BASED FOG ANALYTIC NODE ARCHITECTURE WITH RECONFIGURABLE APPLICATION PLANE

TAN TZE HON

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

MARCH 2022

DEDICATION

Dedicated to

my beloved parents and sponsors

for their support and encouragement.

ACKNOWLEDGEMENT

First of all, I would like to acknowledge everyone who has supported me throughout this research. Particularly, I would like to express my greatest and sincerest appreciation to my supervisor, Associate Professor Ir. Dr. Muhammad Nadzir Bin Marsono and co-supervisor Associate Professor Dr. Ooi Chia Yee. Their continual guidance and motivation lead to the completion of this research. Above all, their enlightenment, inspiration, and support are the reasons I am capable to work on this research.

Besides, I would also like to thank everyone who have supported me directly and indirectly in this research. Specifically, I sincerely thank the reviewers for providing us valuable comments and suggestions to improve our research. I also thank my fellow researchers who helped and accompanied me during this period of time.

Lastly, I would like to thank my supervisor again for everything from the very beginning since 2008 as my mentor and academic advisor until now as my research supervisor. Thank you!

ABSTRACT

Fog computing extends computer networks by embedding analytics into intermediary network devices that are closer to data sources. Fog computing processes sensors data locally, conserves network bandwidth, improves process efficiency, and protects information privacy. However, a fog computing node demands high throughput, low latency, and high energy efficiency in data and packet processing. Besides, a fog node needs high architectural flexibility to enable timely functional updates for maintaining the relevancy of hosted analytics. This thesis proposes a field-programmable gate array (FPGA) based fog node architecture with reconfigurable application plane for fog analytics. The proposed fog node's application plane can be remotely reconfigured at run-time to enable dynamic redeployment of various fog analytics. The reconfigurable application plane allows run-time queuing scheme alteration to prioritize certain network ports and supports scaling on processing entity to cope with increased application loads. The proposed fog node architecture is tested experimentally on the NetFPGA development board with a case study on time-series anomaly detection analytics. The proposed fog node is a monolithic FPGA implementation without general-purpose processor utilization with very low intra-chip communication overhead (less than 24 ns). A customized reconfiguration controller for dynamic partial reconfiguration (DPR) is implemented internally within the FPGA device with 15.34 ms reconfiguration time (i.e., service downtime) at near 3.2 Gbps reconfiguration throughput. Add-on architectures and mechanisms are also proposed to enable service-uninterrupted remote DPR, which are impactful to applications with minimum or no tolerance on service interruption. The implemented FPGA-based time-series anomaly detection analytics have been benchmarked with the Numenta Anomaly Benchmark (NAB) environment for performance assessment. The latency and throughput improvement for the KNN CAD (i.e., a *k*-nearest neighbors based analytics) in FPGA implementation over software is approximately 61×, while Windowed Gaussian (i.e., a Gaussian-based analytics) is 42× and thus, both analytics attained <1 ms real-time responsiveness. The proposed FPGA-based fog node significantly exhibits high energy efficiency, low latency, and high throughput in network packet and data analytics processing for greener fog networks and real-time Internet of Things (IoT) applications.

ABSTRAK

Pengkomputeran kabus meningkatkan keupayaan rangkaian komputer dengan membenamkan analitik ke dalam peranti perantara yang lebih dekat dengan sumber data. Pengkomputeran kabus memproses data dari penderia secara setempat, menjimatkan lebar jalur rangkaian, meningkatkan kecekapan proses, dan melindungi privasi maklumat. Namun, nod pengkomputeran kabus memerlukan daya pemprosesan yang tinggi, kependaman yang rendah, dan kecekapan tenaga yang tinggi dalam pemprosesan data dan paket. Selain itu, nod kabus memerlukan fleksibiliti seni bina yang tinggi untuk membolehkan kemas kini fungsian ketepatan masa untuk mengekalkan relevansi analitik yang dihoskan. Tesis ini mencadangkan seni bina nod kabus berdasarkan tatasusunan get boleh-program medan (FPGA) dengan satah aplikasi yang dapat dikonfigurasi semula untuk analitik kabus. Satah aplikasi nod kabus yang dicadangkan dapat dikonfigurasikan dari jarak jauh pada masa jalan untuk membolehkan kerah tugas semula secara dinamik oleh pelbagai analitik kabus. Satah aplikasi yang dapat dikonfigurasi semula membolehkan perubahan dalam skema baris gilir pada masa jalan untuk memberikan keutamaan kepada port rangkaian tertentu dan menyokong penskalaan pada entiti pemprosesan untuk mengatasi peningkatan beban aplikasi. Seni bina nod kabus yang dicadangkan telah diuji secara eksperimen di papan pembangunan NetFPGA dengan kajian kes dalam analitik pengesanan anomali siri masa. Nod kabus yang dicadangkan adalah implementasi FPGA monolitik tanpa penggunaan pemproses tujuan umum dengan overhed komunikasi dalaman cip yang sangat rendah (kurang dari 24 ns). Pengawal konfigurasi semula yang disesuaikan untuk konfigurasi separa dinamik (DPR) telah diimplementasikan secara dalaman di dalam peranti FPGA dengan masa konfigurasi 15.34 ms (iaitu, waktu henti perkhidmatan) pada daya pemprosesan konfigurasi ulang hampir 3.2 Gbps. Seni bina dan mekanisme tambahan juga dicadangkan untuk membolehkan DPR jarak jauh tanpa gangguan perkhidmatan, yang berpengaruh kepada aplikasi dengan toleransi terendah atau tanpa toleransi terhadap gangguan perkhidmatan. Analitik pengesanan anomali siri masa berasaskan FPGA telah ditanda aras dengan *Numenta Anomaly Benchmark* (NAB) untuk penilaian prestasi. Penambahbaikan kependaman dan peningkatan daya pemprosesan untuk KNN CAD (iaitu, analitik berdasarkan k-jiran terdekat) dalam implementasi FPGA berbanding perisian adalah sekitar 61×, sementara Window Gaussian (iaitu, analitik berasaskan Gaussian) adalah 42× dan dengan itu, kedua-dua analitik mencapai <1 ms respons masa nyata. Nod kabus berasaskan FPGA yang dicadangkan jelas menunjukkan kecekapan tenaga yang tinggi, kependaman yang rendah, dan daya pemprosesan yang tinggi dalam pemprosesan paket rangkaian dan analitik data untuk merealisasikan rangkaian kabus yang lebih hijau dan aplikasi-aplikasi IoT masa nyata.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AANN	-	auto-associative neural network
ALU	-	arithmetic logic unit
ARMA	-	autoregressive moving average
ASIC	-	application-specific integrated circuit
BRAM	-	block random-access memory
CAM	-	content-addressable memory
CLB	-	configurable logic block
COMAC	-	Commercial Aircraft Corporation of China
CPU	-	central processing unit
CSV	-	comma-separated values
DMR	-	dual modular redundancy
DPR	-	dynamic partial reconfiguration
DS	-	Design Suite
DSP	-	digital signal processing
ETH	-	Ethernet
FIFO	-	first-in, first-out
FN	-	false negative
FP	-	false positive
FPGA	-	field-programmable gate array
GbE	-	Gigabit Ethernet
GPP	-	general-purpose processor
GPU	-	graphics processing unit
GUI	-	graphical user interface
HDL	-	hardware description language
HSSIO	-	high-speed serial I/O
HW/SW	-	hardware/software

ICAP	-	Internal Configuration Access Port
IDE	-	integrated development environment
IoT	-	Internet-of-Things
IP	-	intellectual property
ISE	-	Integrated Software Environment
ISP	-	internet service provider
JTAG	-	Joint Test Action Group
LUT	-	lookup table
LSTM	-	long short-term memory
MTU	-	maximum transmission unit
MUX	-	multiplexer
NAB	-	Numenta Anomaly Benchmark
NIC	-	network interface card
NIDS	-	network-based intrusion detection system
NIPS	-	network-based intrusion prevention system
OSI	-	Open Systems Interconnection
QoS	-	quality-of-service
PAR	-	place and route
RISC-V	-	Reduced Instruction Set Computer version 5
PC	-	personal computer
PCIe	-	peripheral component interconnect express
RAM	-	random-access memory
RLE	-	run-length encoding
RNN	-	recurrent neural network
RP	-	reconfigurable partition
SARIMA	-	seasonal autoregressive integrated moving average
SATA	-	serial advanced technology attachment
SEU	-	single event upset
SoC	-	system on a chip

SoPC	-	system on a programmable chip
SRAM	-	static random-access memory
SSD	-	solid-state drive
TCP/IP	-	Transmission Control Protocol over Internet Protocol
UDP/IP	-	User Datagram Protocol over Internet Protocol
XaaS	-	anything-as-a-service
XPS	-	Xilinx Platform Studio

LIST OF SYMBOLS

$\overline{x}$	-	Mean
σ^2	-	Variance
σ	-	Standard deviation

CHAPTER 1

INTRODUCTION

1.1 Recent Trends

The recent emergence of the high-speed network (e.g., 5G) accelerates sensors growth in Internet-of-Things (IoT) by providing the high-performance network infrastructure required to implement a more connected world [3]. Inherently, an unprecedented volume of data is generated by IoT sensors deployed for various applications [4]. To monetize the data, data analytics and machine learning techniques [5–9] are employed to discover its intrinsic information for decision support and automation (e.g., regulating process) purposes [4]. However, the effectiveness and efficiency of conventional analytics diminish as the volume, velocity, veracity, and variety of data increase drastically [10, 11]. Consequently, big data analytics [10–13] are proposed to leverage these huge volumes, fast-growing, and complex data generated from various heterogeneous sources. Fast data analytics [4, 14, 15] has been proposed to focus on the data velocity aspect alone, where the data are highly time-sensitive and require real-time processing (i.e., low latency).

Fog computing [16–18] has emerged as a viable solution for applications with fast data processing, where data analytics are embedded into the network nodes that are placed much closer to the data source compared to the cloud servers. Hence, the analytics in fog nodes can process sensors' data at optimal deployment location depending on the required responsiveness from applications. Moreover, transportation of irrelevant data across the network can be avoided by having the generated data processed locally, which could conserve network bandwidth, improve process efficiency, and safeguard information privacy [16–18]. From the IoT perspective, applications require both fog localization and cloud globalization, where fog nodes could de-emphasize cloud servers' centralized computing environment and provide responses in real-time by eliminating the round-trip latency to cloud servers. Apart from that, fog node scalability allows sensor data to be processed in stages and distributively across the network tiers [16]. A study reported that the mean energy expenses in fog computing are 40.48% lesser than in the cloud [19].

Generally, a fog node can be implemented in several ways, which are application-specific integrated circuit (ASIC) design, software executing in general-purpose processor (GPP), field-programmable gate array (FPGA), and combination of them. ASIC design exhibits very little architectural flexibility and it is not suitable for applications with dynamicity (e.g., fog or IoT applications), where the frequent functional updates are required to maintain its relevancy. Conversely, software approach exhibits highest flexibility from its nature of programmability but it lacks processing throughput and has high processing latency due to instruction stream nature that requires huge memory cycles [20, 21]. The combinations of GPP with accelerators in either FPGA or ASIC are common in HW/SW co-design setups as it can exhibits the balance of processing throughput and architectural flexibility. However, the communication overhead between GPP and accelerators is high and it lacks processing efficiency, which can negatively impact the real-time IoT applications.

To accomplish the objectives of real-time fog analytics in fog computing, fog nodes require the factors as follows that can be satisfied with monolithic FPGA implementation as it exhibits a balance of data processing throughput [22] and architectural flexibility [23, 24]:

1. High computing performance (i.e., high throughput and low latency in data processing) is expected to cope with the growing bandwidth demand in network and the increasing deployed IoT sensors. Meanwhile, low latency computation is needed to meet the real-time data processing demand in IoT applications with fast data. The implementation in FPGA allows pipelined and parallel data processing, which has been commonly adopted in network packet processing [25]. However, the architecture in a design is customized to exploit the computing performance in FPGA implementation optimally.
2. Architectural flexibility is needed by IoT applications for timely functional-update to remain relevant and cope with applications dynamicity, where

operating requirements, protocols, and policy are subjected to alteration from time-to-time. Because of this, newer FPGA devices support dynamic partial reconfiguration (DPR) feature by allowing alteration in regional circuitry at run-time [26] to adapt to the new application operating requirements. Technically, the utilization of DPR feature in FPGA requires a proper design methodology paired with the internal mechanism and reconfigurable architecture to handle the run-time internal circuitry alteration process.

3. High service availability is an important factor in IoT monitoring, data collection, and data processing applications, where service interruptions cause missing data samples that can negatively impact monitoring analytic [27]. Most intermediary devices including fog nodes are always active to preserve end-to-end nodes connectivity and continual fog data processing. The impact of service interruptions is highly significant for mission-critical applications or systems deployed on the gateway between internet service providers (ISPs) [28]. Thus, the reconfigurable architecture in an FPGA-based system that supports smooth remote functional updates with dynamic partial reconfiguration feature is necessary, especially for application that has little to no tolerance toward service interruption.

4. High energy efficiency in both network packet and data analytics processing is one of the critical aspect in fog computing as well. Specifically, the fog nodes near to sensors may have very limited power supply and thus, having high energy efficiency helps to lower the power consumption. High energy efficiency in processing eases the deployment for embedded system in remote area as the devices can be powered from low-power batteries and recharged from solar panel. In data processing, the FPGA implementation exhibits 1.2–22.3× better energy efficiency than the implementation in graphics processing unit (GPU) and GPP [29].

In short, the implementation of fog node with FPGA in monolithic form has various strengths, but it is non-trivial due to the prerequisite of having a good framework, architecture, and mechanism in FPGA design. Additionally, a monolithic FPGA design is not suitable for applications that do not appreciate architectural flexibility and functional updates, where an ASIC design is a better option.

1.2 Problem Statement

Fog computing and network processing applications require processing throughput, real-time latency, and architectural flexibility. Specifically, the processing throughput is needed to meet the growing bandwidth demand [30] from new devices. Real-time latency is essential when the applications utilizing the network infrastructure are time-sensitive and demand responsiveness (e.g., live streaming applications, monitoring applications, and control applications). Besides, the network applications and protocols are dynamic [31], which causes the network devices to require timely functional updates to remain relevant. Since fog computing embeds data analytics into the network devices, its applications share similar requirements as network processing applications.

Specifically, the processing throughput is required to fulfill the growth of new IoT sensors, while architectural flexibility is needed by data analytics to adapt applications dynamicity and data concept drift [32, 33]. The real-time processing latency is critical for fog applications with fast data [4, 14, 15]. FPGA implementation can be considered a feasible solution for such applications due to its low latency and high throughput data processing characteristic paired with datapath reconfigurability, which is critical to real-time and evolving applications. Several works [34–36] have studied and reviewed the suitability and feasibility of employing FPGA in fog and IoT applications. Based on the experiments by Biookaghazadeh et al. [34], the FPGA key advantages for edge computing compared to GPUs are consistent throughput, up to 4$\times$ lower power consumption, up to 30.7$\times$ better energy efficiency, better thermal stability, and lower energy cost per functionality. The benefits of utilizing FPGAs in data (pre)processing near to its source have been demonstrated in [37].

The existing FPGA-based fog node architectures [38–42] are mainly based on HW/SW co-design framework, where the compute-intensive tasks are offloaded to FPGA-based accelerators, while task management is implemented in software and executes in an embedded GPP or a host PC. As a result, these fog node architectures [38–42] are hardly to be applied in gigabit networks or high data rate use cases due to the limited processing capability of GPP on network packet processing. Furthermore, there is a communication overhead between the FPGA-based analytics in reconfigurable fabric and embedded GPP executing the task management routines. The existing FPGA-based network middlebox [43–45] including our previous work [46,47] utilized the DPR feature for timely functional module update. However, existing FPGA-based network middleboxes [43–47] have limited architectural flexibility, which is up to a functional module level that be reconfigured at run-time. Since fog node involves the incorporation of data analytics into a network device, the existing FPGA-based network middlebox requires greater architectural flexibility to host additional fog analytics and its interface. Besides, the architectural flexibility to support DPR on an application plane allows dynamic functional modules allocation based on remote reconfiguration. Specifically, a reconfigurable application plane enables alteration on the internal queuing scheme and scaling on the application processing module at run-time to adapt network prioritization and future increased processing throughput demand.

Service availability is another important factor required by fog computing applications due to these applications are expected to be in service at all times, given the fact that fog nodes are transporting packets containing very critical pieces of sensor data across the networks. Additionally, frequent functional updates on these devices could hinder them from meeting the requirement of high availability. For example, fog computing analytics is retrained frequently to maintain its accuracy. Therefore, service interruptions caused by frequent remote functional updates can be impactful to applications with little to no tolerance (e.g., monitoring applications, data streaming applications, and control applications). Even though the functional update in [48] does not cause service interruption, it is only applicable to a single flow table but not applicable on a functional unit or the data plane. In short, an FPGA-based reconfigurable architecture with mechanisms to enable service-uninterrupted remote functional updates is a requirement for an FPGA-based fog node to support applications without tolerance for service interruption.

In IoT applications, anomaly detection analytics has been commonly employed to monitor time-series data for anomalous events. Existing FPGA-based anomaly detection analytics on time-series often depend on a time-series model [49], e.g., long short-term memory (LSTM) [50, 51] or autoregressive moving average (ARMA) [52] for detecting anomalous events. A software-based anomaly detection analytics with a reliable non-parametric approach [49] (known as KNN CAD) is proposed for anomaly detection in one-dimensional time-series data. This anomaly detection analytics [49] has been benchmarked with Numenta Anomaly Benchmark (NAB) [53, 54] and it is available as open-source at [55]. However, the software implementation [49, 55] limits its processing throughput and exhibits higher latency due to the nature of the instruction stream that requires a number of memory cycles [20, 21]. Hence, the well-established NAB is a good option to be used for the FPGA-based time-series anomaly detection analytics development workbench.

1.3 Research Motivation

Current trends on device connectivity and intelligence imply demands for fog node with high computational performance (high throughput & low latency) and high architectural flexibility. FPGA-based implementation has been a consolidated approach to satisfy applications that require both processing performance and datapath flexibility. Additionally, the DPR feature of FPGA and remote connectivity has made a service-uninterrupted dynamic redeployment possible. Furthermore, the customized datapath in an FPGA design can be defined in software from a remote entity at run-time provided that the underlying reconfigurable architecture supports remote connectivity and DPR.

The development of such versatile fog node is the primary focus of this research work. Employing the ICAP primitive in an FPGA device, DPR can be controlled internally by the controller that is implemented with internal logic resources. Consequently, the FPGA design can be implemented within a single chip, which exhibits high scalability and low device overhead at the system level. Besides, the monolithic FPGA design exhibits low communication between network processing circuitry and data analytics circuitry to reduce overall processing latency and improve energy efficiency.

1.4 Research Objectives

The primary aim of this thesis is to propose a low latency and low communication overhead FPGA-based fog node architecture with reconfigurable application plane to support dynamic redeployment on hosted fog analytics. The objectives of this research work are:

1. To propose, devise, and develop an FPGA-based fog node with reconfigurable application plane, that exhibits a higher degree of architectural flexibility for dynamic redeployment of various fog analytics.
2. To propose, devise, and develop an FPGA-based reconfigurable architecture with supports on service-uninterrupted remote functional updates.
3. To propose, devise, and develop FPGA-based time-series anomaly detection analytics as a case study for the proposed FPGA-based fog node, that has low processing latency to meet the real-time (<1 ms) requirement in fog applications.

1.5 Scope of Work

The scope of this research are:

1. The proposed architectures in this research are tested on the NetFPGA CML development board as this development board can be obtained off-the-shelf and it includes an FPGA device (Xilinx Kintex 7 XC7K325T-1FFG676) that supports the DPR feature. With modularity in the proposed architecture, the migration effort to the other development boards is minimized, where changes are mainly on the peripherals interface.
2. The proposed architectures can handle packets size up to 2048 Bytes, which is larger than the maximum transmission unit (MTU) of Ethernet V2. This limitation can be lifted by increasing the FIFO depth at the cost of BRAM resources in FPGA.

3. The proposed architectures exclude the security aspect in the remote reconfiguration as add-on security modules and mechanisms can be utilized for use cases with security concerns.
4. The proposed architectures are in stand-alone and network-attached (i.e., host PC is not required) mode, which is targeted for single-chip FPGA implementation. The single-chip FPGA implementation exhibits high scalability and low device overhead at the system level.
5. The partial bitstream compression algorithm used in the proposed architectures is 64-bit run-length encoding (RLE).
6. A dual modular redundancy (DMR) approach is adopted to implement the service-uninterrupted remote functional update, where the application modules can cover for each other when either one is being reconfigured dynamically.
7. In the service-uninterrupted remote functional update implementation, the application modules exclude fog analytics to reduce the complexity as it is for proof-of-concept purposes.
8. Network packet processing in the proposed fog node is up to Ethernet frame (i.e., layer 2), which is leveraged from NetFPGA Learning content-addressable memory (CAM) [56] Switch. The network protocol and packet processing can be updated from time-to-time with the remote DPR availability on the proposed fog node's application plane.
9. A time-series anomaly detection analytics IP core is used as a case study for fog computing applications, where the IoT data are mostly available in time-series with its application on monitoring the anomalous events.
10. The employed time-series anomaly detection analytics are adopted from NAB open-source repository [54]. This ensures the benchmarked performance is fair and within the same environment. Besides, the analytics algorithms are available as open-source for public reference.
11. The application IP cores used in case studies are for proof-of-concept purposes, where its complexity and performance are not the primary focus of this work. Complex application IP core may require much higher logic resources amount, which could be fulfilled by migration to newer FPGA device with larger capacity.

1.6 Research Contributions

In this research, the contributions are:

1. An FPGA-based fog node architecture with reconfigurable application plane for fog computing applications. The fog analytics is attached directly to the network processing entity in the pipelined datapath, which reduces the communication overhead between the two processing entities. With reconfigurable application plane, the functional module in the application plane can be dynamically allocated to meet run-time utilization demand, where the queuing schemes can be altered to prioritize certain network ports while the processing entity can be scaled to fulfill increased throughput.
2. An FPGA-based service-uninterrupted remote functional updates mechanism and architecture for applications with high service availability requirement. Service-uninterrupted functional updates mechanism and architecture enable FPGA-based fog node to meet high availability requirements, where these devices are expected to be in service at all times with minimal to no interruption.
3. An FPGA-based time-series anomaly detection analytics architecture as a case study for the proposed fog node. The FPGA-based fog analytics is used as a case study to demonstrate its integration into the proposed fog node and verify their functionality. The FPGA-based fog analytics are able to meet the real-time latency ($<$1 ms) requirement since its datapath can be pipelined and the data can be processed in parallel.

1.7 Thesis Organization

The remaining content of the thesis is structured as follows:

- Chapter 2 describes the theoretical background and related works. The theoretical background includes fog computing, network processing, FPGA suitability for fog computing, and NetFPGA. The related work covers FPGA-based fog node, FPGA-based service-uninterrupted remote functional updates, and FPGA-based anomaly detection analytics.

- Chapter 3 describes the procedure of this research towards achieving the objectives. First, an overview of the proposed FPGA-based fog node and its high-level architecture is discussed. Then, the development setup of the proposed FPGA-based fog node architecture is provided together with its development tools and environment setup. This chapter also discusses the verification and validation methods used.
- Chapter 4 presents the implementation and development of the proposed FPGA-based fog node with the NetFPGA CML development board. The evaluation of the developed fog node is provided for verification and benchmark purposes.
- Chapter 5 presents the FPGA-based add-ons architectures and mechanisms to enable remote DPR without causing service interruption. These add-on architectures are significant to applications that sensitive toward service interruption, where the application services circuitry implemented in FPGA is halted during the DPR period.
- Chapter 6 presents the implementation and development of two different FPGA-based anomaly detection analytics and their integration into the proposed FPGA-based fog node as a case study. The developed anomaly detection analytics are benchmarked with NAB [53, 54] for verification and analysis purposes.
- Chapter 7 summarizes the presented content and reemphasizes the thesis contributions. The potential future works are suggested in the last section of this chapter.

REFERENCES

1. Delen, D. and Ram, S. Research challenges and opportunities in business analytics. *Journal of Business Analytics*, 2018. 1(1): 2–12.
2. Aggarwal, C. C. *Outlier Analysis, Second Edition*. Springer. 2016. ISBN 9783319837727.
3. IEEE. 3 Key Benefits of 5G. URL `https://innovationatwork.ieee.org/3-key-benefits-of-5g/`, accessed Feb 20, 2022.
4. Mohammadi, M., Al-Fuqaha, A., Sorour, S. and Guizani, M. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, 2018. 20(4): 2923–2960.
5. Chen, M., Challita, U., Saad, W., Yin, C. and Debbah, M. Artificial neural networks-based machine learning for wireless networks: A tutorial. *IEEE Communications Surveys & Tutorials*, 2019. 21(4): 3039–3071.
6. Bansal, A., Sharma, M. and Goel, S. Improved K-mean clustering algorithm for prediction analysis using classification technique in data mining. *International Journal of Computer Applications*, 2017. 157(6): 35–40.
7. Petneházi, G. Recurrent neural networks for time series forecasting. *arXiv preprint arXiv:1901.00069*, 2019.
8. Giorgio, A., Rizzi, M. and Guaragnella, C. Efficient detection of ventricular late potentials on ECG signals based on wavelet denoising and SVM classification. *Information*, 2019. 10(11): 1–18.
9. Feng, W., Zhu, Q., Zhuang, J. and Yu, S. An expert recommendation algorithm based on Pearson correlation coefficient and FP-growth. *Cluster Computing*, 2019. 22(3): 7401–7412.
10. Russom, P. Big data analytics. *TDWI best practices report (fourth quarter 2011)*, 2011. 19(4): 1–34.
11. Davenport, T. H., Barth, P. and Bean, R. How ’big data’ is different. *MIT Sloan Management Review*, 2012. 54(1): 22–24.

12. Tsai, C.-W., Lai, C.-F., Chao, H.-C. and Vasilakos, A. V. Big data analytics: a survey. *Journal of Big data*, 2015. 2(1): 1–32.

13. Raghavan, R. and Perera, D. G. A fast and scalable FPGA-based parallel processing architecture for K-means clustering for big data analysis. *Proceedings of the 2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM).* Victoria, BC, Canada. 2017. 1–8.

14. Miloslavskaya, N. and Tolstoy, A. Big data, fast data and data lake concepts. *Procedia Computer Science*, 2016. 88: 300–305.

15. Betts, R. and Hugg, J. *Fast Data: Smart and at Scale*. O'Reilly Media, Incorporated. 2015. ISBN 9781491940372.

16. CISCO. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. URL `https://www.iotjournaal.nl/wp-content/uploads/2016/08/computing-overview.pdf`, accessed Feb 20, 2022.

17. Yi, S., Hao, Z., Qin, Z. and Li, Q. Fog computing: Platform and applications. *Proceedings of the 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb).* Washington, DC, USA. 2015. 73–78.

18. Dastjerdi, A. V. and Buyya, R. Fog Computing: Helping the Internet of Things Realize Its Potential. *Computer*, 2016. 49(8): 112–116.

19. Sarkar, S. and Misra, S. Theoretical modelling of fog computing: A green computing paradigm to support IoT applications. *IET Networks*, 2016. 5(2): 23–29.

20. Carson, E., Demmel, J., Grigori, L., Knight, N., Koanantakool, P., Schwartz, O. and Simhadri, H. V. Write-avoiding algorithms. *Proceedings of the 2016 IEEE International Parallel and Distributed Processing Symposium.* Chicago, IL, USA. 2016. 648–658.

21. Becker, J. and Hartenstein, R. Configware and morphware going mainstream. *Journal of Systems Architecture*, 2003. 49(4): 127–142.

22. Beneš, T., Bartík, M. and Kubalík, P. High throughput and low latency LZ4 compressor on FPGA. *Proceedings of the 2019 International Conference*

on ReConFigurable Computing and FPGAs (ReConFig). Cancun, Mexico. 2019. 1–5.

23. Pijetlovic, S., Subotić, M., Marinkovic, V. and Pjevalica, N. FIR filter implementation for high-performance application in a high-end FPGA. *Proceedings of the 2018 26th Telecommunications Forum (TELFOR)*. Belgrade, Serbia. 2018. 1–4.

24. Lightbody, G., Browne, F. and Haberland, V. Custom hardware versus cloud computing in big data. In: Schuster, A. J., ed. *Understanding Information*. Berlin, Germany: Springer. 175–193. 2017.

25. Naous, J., Gibb, G., Bolouki, S. and McKeown, N. NetFPGA: reusable router architecture for experimental research. *Proceedings of the 2008 ACM workshop on Programmable routers for extensible services of tomorrow*. Seattle, WA, USA. 2008. 1–7.

26. Xilinx. UG702 (v14.5) Partial Reconfiguration User Guide. *Xilinx Inc., Apr*, 2013. URL `https://japan.xilinx.com/support/documentation/sw_manuals_j/xilinx14_6/ug702.pdf`, accessed Feb 20, 2022.

27. Brown, M. L. and Kros, J. F. Data mining and the impact of missing data. *Industrial Management & Data Systems*, 2003. 103(8): 174–198.

28. Katayama, M., Kai, H., Yoshida, J., Inami, M., Yamada, H., Shiomoto, K. and Yamanaka, N. A 10 Gb/s firewall system for network security in photonic era. *IEICE transactions on communications*, 2005. 88(5): 1914–1920.

29. Qasaimeh, M., Denolf, K., Lo, J., Vissers, K., Zambreno, J. and Jones, P. H. Comparing energy efficiency of CPU, GPU and FPGA implementations for vision kernels. *Proceedings of the 2019 IEEE international conference on embedded software and systems (ICESS)*. Las Vegas, NV, USA. 2019. 1–8.

30. Kocovic, P. Four laws for today and tomorrow. *Journal of Applied Research and Technology*, 2008. 6(3): 133–146.

31. ARISTA. Four key trends in the networked use of FPGAs, 2018. URL `https://www.arista.com/assets/data/pdf/Whitepapers/Trends-in-FPGA-WP.pdf`, accessed Feb 20, 2022.

32. Tian, X., Sun, Q., Huang, X. and Ma, Y. A dynamic online traffic classification methodology based on data stream mining. *Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering*. Los Angeles, CA, USA. 2009, vol. 1. 298–302.

33. Mulinka, P. and Casas, P. Stream-based machine learning for network security and anomaly detection. *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*. Budapest, Hungary. 2018. 1–7.

34. Biookaghazadeh, S., Zhao, M. and Ren, F. Are FPGAs suitable for edge computing? *Proceedings of the 2018 USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*. Boston, MA, USA. 2018. 1–6.

35. Liu, Y., Fieldsend, J. E. and Min, G. A framework of fog computing: Architecture, challenges, and optimization. *IEEE Access*, 2017. 5: 25445–25454.

36. De La Piedra, A., Braeken, A. and Touhafi, A. Sensor systems based on FPGAs and their applications: A survey. *Sensors*, 2012. 12(9): 12235–12264.

37. Woods, L. and Alonso, G. Fast data analytics with FPGAs. *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering Workshops*. Hannover, Germany. 2011. 296–299.

38. Tarasov, I. and Potekhin, D. FPGA-Based SOC Architecture for Fog and Edge Computing Applications. *Proceedings of the 2020 International Conference on High-Performance Computing Systems and Technologies in Scientific Research, Automation of Control and Production*. Barnaul, Russia. 2020. 3–13.

39. Jiang, S., He, D., Yang, C., Xu, C., Luo, G., Chen, Y., Liu, Y. and Jiang, J. Accelerating mobile applications at the network edge with software-programmable FPGAs. *Proceedings of the 2018 IEEE Conference on Computer Communications (INFOCOM)*. Honolulu, HI, USA. 2018. 55–62.

40. Rodríguez, A., Valverde, J., Portilla, J., Otero, A., Riesgo, T. and De la Torre, E. FPGA-Based high-performance embedded systems for adaptive edge

computing in cyber-physical systems: The ARTICo3 framework. *Sensors*, 2018. 18(6): 1–30.

41. Barbieri, S., Casasopra, F., Brondolin, R. and Santambrogio, M. D. FARD: accelerating distributed fog computing workloads through embedded FPGAs. *ACM SIGBED Review*, 2020. 17(1): 56–62.

42. Gomes, T., Pinto, S., Tavares, A. and Cabral, J. Towards an FPGA-based edge device for the Internet of Things. *Proceedings of the 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*. Luxembourg, Luxembourg. 2015. 1–4.

43. Hager, S., Bendyk, D. and Scheuermann, B. Partial reconfiguration and specialized circuitry for flexible FPGA-based packet processing. *Proceedings of the 2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. Cancun, Mexico. 2015. 1–6.

44. Yin, D., Unnikrishnan, D., Liao, Y., Gao, L. and Tessier, R. Customizing virtual networks with partial FPGA reconfiguration. *ACM SIGCOMM Computer Communication Review*, 2011. 41(1): 125–132.

45. Mishra, V., Chen, Q. and Zervas, G. REoN: A protocol for reliable software-defined FPGA partial reconfiguration over network. *Proceedings of the 2016 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. Cancun, Mexico. 2016. 1–7.

46. Tan, T. H., Ooi, C. Y. and Marsono, M. N. rrBox: A remote dynamically reconfigurable network processing middlebox. *Proceedings of the 2015 25th International Conference on Field Programmable Logic and Applications (FPL)*. London, United Kingdom. 2015. 1–4.

47. Tan, T. H., Ooi, C. Y. and Marsono, M. N. *Remote Dynamically Reconfigurable Platform for Network Processing Application on NetFPGA*. Master thesis. Universiti Teknologi Malaysia. 2015.

48. Zhou, S., Jiang, W. and Prasanna, V. K. A flexible and scalable high-performance OpenFlow switch on heterogeneous SoC platforms. *Proceedings of the 2014 IEEE International Performance Computing and Communications Conference (IPCCC)*. Austin, TX, USA. 2014. 1–8.

49. Burnaev, E. and Ishimtsev, V. Conformalized density-and distance-based anomaly detection in time-series data. *arXiv preprint arXiv:1608.04585*, 2016.

50. Que, Z., Liu, Y., Guo, C., Niu, X., Zhu, Y. and Luk, W. Real-time Anomaly Detection for Flight Testing using AutoEncoder and LSTM. *Proceedings of the 2019 International Conference on Field-Programmable Technology (ICFPT)*. Tianjin, China. 2019. 379–382.

51. Wu, D., Sun, Z., Zhu, Y., Tian, L., Zhu, H., Xiong, P., Cao, Z., Wang, M., Zheng, Y., Xiong, C., Jiang, H., Tsoi, K. H., Niu, X., Mao, W., Feng, C., Zha, X., Deng, G. and Luk, W. Custom machine learning architectures: towards realtime anomaly detection for flight testing. *Proceedings of the 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. Vancouver, BC, Canada. 2018. 1323–1330.

52. MacEachern, L. and Vazhbakht, G. Configurable FPGA-Based Outlier Detection for Time Series Data. *Proceedings of the 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*. Springfield, MA, USA. 2020. 142–145.

53. Lavin, A. and Ahmad, S. Evaluating Real-Time anomaly detection algorithms–The Numenta anomaly benchmark. *Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. Miami, FL, USA. 2015. 38–44.

54. Numenta. The Numenta Anomaly Benchmark on GitHub. URL `https://github.com/numenta/NAB/`, accessed Feb 20, 2022.

55. Burnaev, E. and Ishimtsev, V. KNN-CAD Python implementation on GitHub. URL `https://github.com/numenta/NAB/tree/master/nab/detectors/knncad`, accessed Feb 20, 2022.

56. Kay, N. S. and Marsono, M. Ternary content addressable memory for longest prefix matching based on random access memory on field programmable gate array. *Telkomnika*, 2019. 17(4): 1882–1889.

57. Giladi, R. *Network processors: architecture, programming, and implementation*. Morgan Kaufmann. 2008. ISBN 9780123708915.

58. Yi, S., Li, C. and Li, Q. A survey of fog computing: concepts, applications and issues. *Proceedings of the 2015 Workshop on Mobile Big Data*. Hangzhou, China. 2015. 37–42.

59. Naha, R. K., Garg, S. and Chan, A. Fog computing architecture: Survey and challenges. *arXiv preprint arXiv:1811.09047*, 2018.

60. Dastjerdi, A. V., Gupta, H., Calheiros, R. N., Ghosh, S. K. and Buyya, R. Fog computing: Principles, architectures, and applications. *arXiv preprint arXiv:1601.02752*, 2016.

61. Carson, E., Demmel, J., Grigori, L., Knight, N., Koanantakool, P., Schwartz, O. and Simhadri, H. V. Write-avoiding algorithms. *Proceedings of the 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. Chicago, IL, USA. 2016. 648–658.

62. Bobda, C. *Introduction to Reconfigurable Computing: Architectures, algorithms and applications*. Springer. 2007. ISBN 9781402060885.

63. García, G. J., Jara, C. A., Pomares, J., Alabdo, A., Poggi, L. M. and Torres, F. A survey on FPGA-based sensor systems: towards intelligent and reconfigurable low-power sensors for computer vision, control and signal processing. *Sensors*, 2014. 14(4): 6247–6278.

64. Hervás, M., Alsina-Pagès, R. M., Alías, F. and Salvador, M. An FPGA-based WASN for remote real-time monitoring of endangered species: A case study on the birdsong recognition of botaurus stellaris. *Sensors*, 2017. 17(6): 1331–1356.

65. Tang, B., Chen, Z., Hefferman, G., Pei, S., Wei, T., He, H. and Yang, Q. Incorporating intelligence in fog computing for big data analysis in smart cities. *IEEE Transactions on Industrial informatics*, 2017. 13(5): 2140–2150.

66. Myint, C. Z., Gopal, L. and Aung, Y. L. Reconfigurable smart water quality monitoring system in IoT environment. *Proceedings of the 2017 IEEE/ACIS 16th international conference on computer and information science (ICIS)*. Wuhan, China. 2017. 435–440.

67. Steffen, D., Bai, Y., Bodlak, M., Frolov, V., Huber, S., Jary, V., Konorov, I., Levit, D., Novy, J., Subrt, O. and Virius, M. Overview and Future

Developments of the intelligent FPGA-based DAQ (iFDAQ) of COMPASS. *Proceedings of 38th International Conference on High Energy Physics — PoS(ICHEP2016)*. Chicago, USA. 2017, vol. 282. 912–916.

68. Rajasekaran, C., Jeyabharath, R. and Veena, P. FPGA SoC based multichannel data acquisition system with network control module. *Circuits and Systems*, 2017. 8(2): 53–75.

69. Moore, A. W. NetFPGA. URL `http://netfpga.org/`, accessed Feb 20, 2022.

70. Schallenberg, A. *Dynamic partial self-reconfiguration: Quick modeling, simulation, and synthesis*. Suedwestdeutscher Verlag fuer Hochschulschriften. 2010. ISBN 9783838122632.

71. Engelmann, C., Ong, H. H. and Scott, S. L. The case for modular redundancy in large-scale high performance computing systems. *Proceedings of the 8th IASTED international conference on parallel and distributed computing and networks (PDCN)*. Innsbruck, Austria. 2009. 189–194.

72. Mallavarapu, P., Upadhyay, H. N., Rajkumar, G. and Elamaran, V. Fault-tolerant digital filters on FPGA using hardware redundancy techniques. *Proceedings of the 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*. Coimbatore, India. 2017, vol. 2. 256–259.

73. Legat, U., Biasizzo, A. and Novak, F. SEU recovery mechanism for SRAM-based FPGAs. *IEEE Transactions on Nuclear Science*, 2012. 59(5): 2562–2571.

74. Matsuo, I. B. M., Zhao, L. and Lee, W.-J. A dual modular redundancy scheme for CPU–FPGA platform-based systems. *IEEE Transactions on Industry Applications*, 2018. 54(6): 5621–5629.

75. Zazo, J. F., Lopez-Buedo, S., Sutter, G. and Aracil, J. Automated synthesis of FPGA-based packet filters for 100 Gbps network monitoring applications. *Proceedings of the 2016 ReConFigurable Computing and FPGAs (ReConFig), 2016 International Conference on*. Cancun, Mexico. 2016. 1–6.

76. Kotlar, M., Bojić, D., Punt, M. and Milutinović, V. Survey of deployment locations and underlying hardware architectures for contemporary deep neural networks. *International Journal of Distributed Sensor Networks*, 2019. 15(8): 1–10.

77. Kotlar, M., Bojic, D., Punt, M. and Milutinovic, V. A survey of deep neural networks: Deployment location and underlying hardware. *Proceedings of the 2018 14th Symposium on Neural Networks and Applications (NEUREL)*. Belgrade, Serbia. 2018. 1–6.

78. Byers, C. C. and Wetterwald, P. Fog computing distributing data and intelligence for resiliency and scale necessary for IoT: The internet of things (ubiquity symposium). *Ubiquity*, 2015. 2015(November): 1–12.

79. Weerasinghe, J., Polig, R., Abel, F. and Hagleitner, C. Network-attached FPGAs for data center applications. *Proceedings of the 2016 International Conference on Field-Programmable Technology (FPT)*. Xi'an, China. 2016. 36–43.

80. Khazraee, M., Forencich, A., Papen, G., Snoeren, A. C. and Schulman, A. Shire: Making FPGA-accelerated Middlebox Development More Pleasant. *arXiv preprint arXiv:2201.08978*, 2022.

81. Cook, A. A., Mısırlı, G. and Fan, Z. Anomaly detection for IoT time-series data: A survey. *IEEE Internet of Things Journal*, 2019. 7(7): 6481–6494.

82. Giannoni, F., Mancini, M. and Marinelli, F. Anomaly detection models for IoT time series data. *arXiv preprint arXiv:1812.00890*, 2018.

83. Żabiński, T., Hajduk, Z., Kluska, J. and Gniewek, L. FPGA-Embedded Anomaly Detection System for Milling Process. *IEEE Access*, 2021. 9: 124059–124069.

84. Zhang, X., Kim, J., Lin, Q., Lim, K., Kanaujia, S. O., Xu, Y., Jamieson, K., Albarghouthi, A., Qin, S., Freedman, M. J. *et al.* Cross-dataset time series anomaly detection for cloud systems. *Proceedings of the 2019 USENIX Annual Technical Conference (USENIX ATC 19)*. Renton, WA, USA. 2019. 1063–1076.

85. Laptev, N., Amizadeh, S. and Flint, I. Generic and scalable framework for automated time-series anomaly detection. *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. Sydney, Australia. 2015. 1939–1947.

86. Takashina, Y. Windowed Gaussian anomaly detector Python implementation on GitHub. URL `https://github.com/numenta/NAB/tree/master/nab/detectors/gaussian`, accessed Feb 20, 2022.

87. Ahmad, S., Lavin, A., Purdy, S. and Agha, Z. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 2017. 262: 134–147.

88. Skyline. URL `https://earthgecko-skyline.readthedocs.io/en/latest/`, accessed Feb 20, 2022.

89. Guha, S., Mishra, N., Roy, G. and Schrijvers, O. Robust random cut forest based anomaly detection on streams. *Proceedings of the 2016 International conference on machine learning*. New York City, NY, USA. 2016. 2712–2721.

90. Schneider, M., Ertel, W. and Ramos, F. Expected similarity estimation for large-scale batch and streaming anomaly detection. *Machine Learning*, 2016. 105(3): 305–333.

91. Procaccianti, G., Vetro, A., Ardito, L. and Morisio, M. Profiling power consumption on desktop computer systems. *Proceedings of the 2011 International conference on information and communication on technology*. Toulouse, France. 2011. 110–123.

92. Nezami, K. G., Stephens, P. W. and Walker, S. D. Handel-C Implementation of Early-Access Partial-Reconfiguration for Software Defined Radio. *Proceedings of the 2008 IEEE Wireless Communications and Networking Conference*. Las Vegas, NV, USA. 2008. 1103–1108.

93. Xilinx. XAPP290 (v1.0) Two Flows for Partial Reconfiguration: Module Based or Small Bit Manipulations. *Xilinx Inc., May*, 2002. URL `https://uweb.engr.arizona.edu/~ece506/readings/project-reading/1-partial-reconfiguration/Module%20Based%20Partial%20Reconfiguration%20Xilinx.pdf`, accessed Feb 20, 2022.

94. Lee, W., Noh, K., Kim, S. and Heo, J. Efficient cooperative transmission for wireless 3D HD video transmission in 60GHz channel. *IEEE Transactions on Consumer Electronics*, 2010. 56(4): 2481–2488.

95. Nasir, A. A., Tuan, H. D., Duong, T. Q. and Debbah, M. NOMA throughput and energy efficiency in energy harvesting enabled networks. *IEEE Transactions on Communications*, 2019. 67(9): 6499–6511.

96. Cavalcante, R. L., Stanczak, S., Schubert, M., Eisenblaetter, A. and Tuerke, U. Toward energy-efficient 5G wireless communications technologies: Tools for decoupling the scaling of networks from the growth of operating power. *IEEE Signal Processing Magazine*, 2014. 31(6): 24–34.

97. Xilinx. UG474 (v1.8) 7 Series FPGAs Configurable Logic Block. *Xilinx Inc., Sep*, 2016. URL `https://www.xilinx.com/support/documentation/user_guides/ug474_7Series_CLB.pdf`, accessed Feb 20, 2022.

98. Nabina, A. and Nunez-Yanez, J. L. Dynamic reconfiguration optimisation with streaming data decompression. *Proceedings of the 2010 International Conference on Field Programmable Logic and Applications*. Milan, Italy. 2010. 602–607.

99. Liu, S., Pittman, R. N., Forin, A. and Gaudiot, J.-L. Minimizing the runtime partial reconfiguration overheads in reconfigurable systems. *The Journal of Supercomputing*, 2012. 61(3): 894–911.

100. Vipin, K. and Fahmy, S. A. A high speed open source controller for FPGA partial reconfiguration. *Proceedings of the 2012 International Conference on Field-Programmable Technology (FPT)*. Seoul, Korea. 2012. 61–66.

101. Joseph, J. M., Mey, M., Ehlers, K., Blochwitz, C., Winker, T. and Pionteck, T. Design space exploration for a hardware-accelerated embedded real-time pose estimation using Vivado HLS. *Proceedings of the 2017 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. Cancun, Mexico. 2017. 1–8.

102. Anirudh, B. K., Venkatraman, V., Kumar, A. R. and Sumam David, S. Accelerating real-time computer vision applications using HW/SW co-

design. *Proceedings of the 2017 International Conference on Computer, Communications and Electronics (Comptelix)*. Jaipur, India. 2017. 458–463.

103. Gangwar, P., Maurya, S., Garg, S., Goyal, S., Kumar, A. S., Dalmia, P. and Pandey, N. Hardware/Software Co-Design of a High-Speed Othello Solver. *Proceedings of the 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*. Dallas, TX, USA. 2019. 1223–1226.

104. Brown, M. L. and Kros, J. F. Data mining and the impact of missing data. *Industrial Management & Data Systems*, 2003. 103(8): 521–611.

105. Metcalfe, A., Green, D., Greenfield, T., Mansor, M., Smith, A. and Tuke, J. *Statistics in Engineering: With Examples in MATLAB® and R, Second Edition*. Chapman and Hall/CRC. 2019. ISBN 9781439895474.

106. Steinberg, W. J. and Price, M. *Statistics alive!* Sage Publications. 2020. ISBN 9781412956574.

107. Korat, U. A., Yadav, P. and Shah, H. An efficient hardware implementation of vector-based odd-even merge sorting. *Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. New York, NY, USA. 2017. 654–657.

108. Vucha, M. and Rajawat, A. Design and FPGA implementation of systolic array architecture for matrix multiplication. *International Journal of Computer Applications*, 2011. 26(3): 18–22.

109. Asgari, B., Hadidi, R. and Kim, H. Proposing a fast and scalable systolic array for matrix multiplication. *Proceedings of the 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. Fayetteville, AR, USA. 2020. 204–204.

110. Lei, J., Yang, G., Xie, W., Li, Y. and Jia, X. A Low-Complexity Hyperspectral Anomaly Detection Algorithm and Its FPGA Implementation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2020. 14: 907–921.

LIST OF PUBLICATIONS

Indexed Journal (SCOPUS)

1. Tze Hon Tan, Chia Yee Ooi, and M. Nadzir Marsono. RtFog: A Real-time FPGA-based Fog Node with Remote Dynamically Reconfgurable Application Plane for Fog Analytics Redeployment. *IEEE Transactions on Green Communications and Networking (TGCN)*. 2022. 6(1), 341–351.
2. Tze Hon Tan, Chia Yee Ooi, and M. Nadzir Marsono. An FPGA-based Network System with Service-uninterrupted Remote Functional Update. *International Journal of Electrical & Computer Engineering (IJECE)*. 2021. 11(4), 3222–3228.
3. Tze Hon Tan, Chia Yee Ooi, and M. Nadzir Marsono. A Customized Reconfiguration Controller with Remote Direct ICAP Access for Dynamically Reconfigurable Platform. *Telecommunication Computing Electronics and Control (TELKOMNIKA)*. 2017. 15(2), 570–577.

Indexed conference proceedings

1. Tze Hon Tan, Chia Yee Ooi, and M. Nadzir Marsono. An FPGA-based Middlebox with Remote Dynamically Reconfigurable Application Plane. *In 2021 IEEE Region 10 Conference (TENCON)*. Auckland, New Zealand. 2021. 52–56.
2. Tze Hon Tan, Chia Yee Ooi, and M. Nadzir Marsono. hpFog: A FPGA-Based Fog Computing Platform. *In 2017 International Conference on Networking, Architecture, and Storage (NAS)*. Shenzhen, China. 2017. 1–2.
3. Tze Hon Tan, Chia Yee Ooi, and M. Nadzir Marsono. A Modular Architecture for Dynamically Reconfigurable Middlebox with Customized Reconfiguration Handler. *In 2016 International Conference on Field Programmable Technology (FPT)*. Xi'an, China. 2016. 221–224.