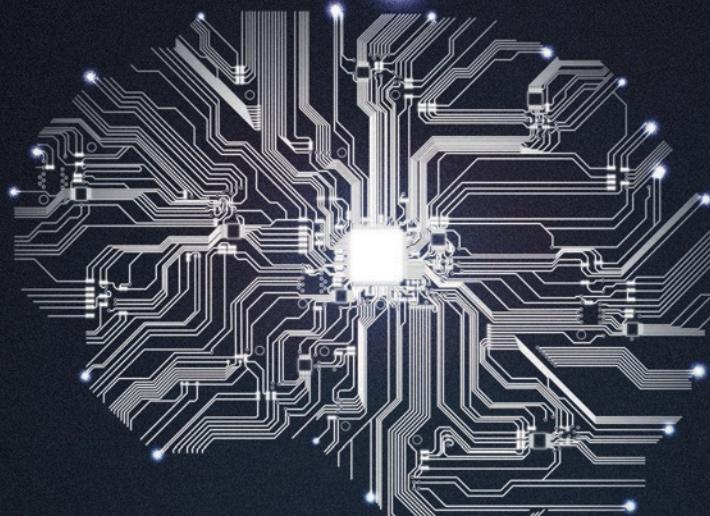


Volume 19, Number 4
Fourth Quarter 2019
ISSN 1531-636X

IEEE Circuits and Systems Magazine



- 6** From Behavioral Design of Memristive Circuits and Systems to Physical Implementations
- 19** Applications of Deep Learning to Audio Generation
- 39** Recent Development in Public Transport Network Analysis From the Complex Network Perspective





Bright Minds. Bright Ideas.



Introducing IEEE Collabratec™

The premier networking and collaboration site for technology professionals around the world.

IEEE Collabratec is a new, integrated online community where IEEE members, researchers, authors, and technology professionals with similar fields of interest can **network** and **collaborate**, as well as **create** and manage content.

Featuring a suite of powerful online networking and collaboration tools, IEEE Collabratec allows you to connect according to geographic location, technical interests, or career pursuits.

You can also create and share a professional identity that showcases key accomplishments and participate in groups focused around mutual interests, actively learning from and contributing to knowledgeable communities. All in one place!

Network.
Collaborate.
Create.

Learn about IEEE Collabratec at
ieee-collabratec.ieee.org



IEEE Circuits and Systems Magazine

Volume 19, Number 4 Fourth Quarter 2019

Features



©ISTOCKPHOTO.COM/GO-UN LEE



IEEE Circuits and Systems Magazine (ISSN 1531-636X) (ICDMEN) is published quarterly by the Institute of Electrical and Electronics Engineers, Inc. Headquarters: 3 Park Avenue, 17th Floor, New York, NY, 10016-5979 USA. Responsibility for the contents rests upon the authors and not upon the IEEE, the Society, or its members. IEEE Service Center (for orders, subscriptions, address changes): 445 Hoes Lane, Piscataway, NJ 08854 USA. Telephone: +1 732 981 0060, +1 800 678 4333. Individual copies: IEEE members US\$20.00 (first copy only), nonmembers US\$176 per copy; US\$7.00 per member per year (includes Society fee) for each member of the IEEE Circuits and Systems Society. Subscription rates available upon request. **Copyright and Reprint Permission:** Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of the U.S. Copyright law for private use of patrons: 1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923; and 2) pre-1978 articles without fee. For other copying, reprint, or republication permission, write to: Copyrights and Permissions Department, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854 USA. Copyright © 2019 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Periodicals postage paid at New York, NY, and at additional mailing offices. **Postmaster:** Send address changes to IEEE Circuits and Systems Magazine, IEEE Operations Center, 445 Hoes Lane, Piscataway, NJ, 08854 USA. Printed in U.S.A.

Digital Object Identifier 10.1109/MCAS.2019.2945207

6 From Behavioral Design of Memristive Circuits and Systems to Physical Implementations

Nima TaheriNejad and David Radakovits

Since Hewlett Packard (HP) announced the passive fabrication of their memristors, various memristive technologies—as a promising emerging technology—have gained ever-increasing attention from the researchers. Although a natural application is using them as memory units, there have been several works in the literature showing their utilization in circuits and systems. While research on various aspects of memristive circuit and systems has been proliferating, the majority of these works are based on simulations at different levels of modeling abstraction. Simulation is a very helpful design tool, and there have been several efforts in modeling memristors; however, we contend that at this point these simulations represent the reality of the behavior of memristors, especially in a circuit or system set-up, only to a very limited extent. We show how this negatively affects the reproduction of designed circuits and systems in different simulation levels, and more importantly in a real-world setup with physical implementation. Following that, we look into some considerations which can improve the reproducibility of the circuit and systems to be designed in the future. We conclude the paper by suggesting certain approaches to tackle these practical challenges at device level as well as circuit and system level.

19 Applications of Deep Learning to Audio Generation

Yuanjun Zhao, Xianjun Xia, and Roberto Togneri

In recent years, deep learning based machine learning systems have demonstrated remarkable success for a wide range of learning tasks in multiple domains such as computer vision, speech recognition and other pattern recognition based applications. The purpose of this article is to contribute a timely review and introduction of state-of-the-art deep learning techniques and their effectiveness in speech/acoustic signal processing. Thorough investigations of various deep learning architectures are provided under the categories of discriminative and generative algorithms, including the up-to-date Generative Adversarial Networks (GANs) as an integrated model. A comprehensive overview of applications in audio generation is highlighted. Based on understandings from these approaches, we discuss how deep learning methods can benefit the field of speech/acoustic signal synthesis and the potential issues that need to be addressed for prospective real-world scenarios. We hope this survey provides a valuable reference for practitioners seeking to innovate in the usage of deep learning approaches for speech/acoustic signal generation.

39 Recent Development in Public Transport Network Analysis From the Complex Network Perspective

Tanuja Shanmukhappa, Ivan Wang-Hei Ho, Chi K. Tse, and Kin K. Leung

A graph, comprising a set of nodes connected by edges, is one of the simplest yet remarkably useful mathematical structures for the analysis of real-world complex systems. Network theory, being an application-based extension of graph theory, has been applied to a wide variety of real world systems involving complex interconnection of subsystems. The application of network theory has permitted in-depth understanding of connectivity, topologies, and operations of many practical networked systems as well as the roles that various parameters play in determining the performance of such systems. In the field of transportation networks, however, the use of graph theory has been relatively much less explored, and this motivates us to bring together the recent development in the field of public transport analysis from a graph theoretic perspective. In this paper, we focus on ground transportation, and in particular the bus transport network (BTN) and metro transport network (MTN), since the two types of networks are widely used by the public and their performances have significant impact to people's life. In the course of our analysis, various network parameters are introduced to probe into the impact of topologies and their relative merits and demerits in transportation. The various local and global properties evaluated as part of the topological analysis provide a common platform to comprehend and decipher the inherent network features that are partly encoded in their topological properties. Overall, this paper gives a detailed exposition of recent development in the use of graph theory in public transport network analysis, and summarizes the key results that offer important insights for government agencies and public transport system operators to plan, design, and optimize future public transport networks in order to achieve more efficient and robust services.

3 Recognitions

4 CAS Society News

Editor-in-Chief

Chai Wah Wu
T.J. Watson Research Center
IBM, USA
E-mail: chaiwahwu@ieee.org

Deputy Editor-in-Chief

Alyssa B. Apsel
Cornell University, USA

Associate Editors

Pamela Abshire
University of Maryland, USA
Ming Cao
University of Groningen, The Netherlands
Herbert Iu
The University of Western Australia, Australia
Ljupco Kocarev
University "Sv. Kiril i Metodij," Macedonia
University of California, San Diego, USA
Anthony Kopa
The Charles Stark Draper Laboratory, USA
Henry Leung
University of Calgary, Canada
Jinhui Lü
Chinese Academy of Sciences, China
Marvin Onabajo
Northeastern University, USA
Sule Ozve
Arizona State University, USA
Xuan Zhang
Washington University in St. Louis, USA

IEEE Periodicals Editorial & Production Services

Editorial/Production Associate — Heather Hilton
Senior Managing Editor — Geraldine Krolin-Taylor
Senior Art Director — Janet Duder
Associate Art Director — Gail A. Schnitzer
Production Coordinator — Theresa L. Smith
Advertising Production Mgr. — Felicia Spagnoli
Dir. Production Services — Peter M. Tuohy
Dir. Editorial Services — Kevin Lisankie
Staff Director, Publishing Operations — Dawn M. Melley
Director, Business Development – Media & Advertising — Mark David

Magazine Deadlines

Final materials for *IEEE Circuits and Systems Magazine* must be received by the Editor on the following dates:

| Issue | Due Date |
|----------------|--------------|
| First Quarter | December 23 |
| Second Quarter | April 1 |
| Third Quarter | July 1 |
| Fourth Quarter | September 15 |

IEEE prohibits discrimination, harassment, and bullying. For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

Digital Object Identifier 10.1109/MCAS.2019.2945432

Scope: Insofar as the technical articles presented in the proposed magazine, the plan is to cover the subject areas represented by the Society's transactions, including: analog, passive, switch capacitor, and digital filters; electronic circuits, networks, graph theory, and RF communication circuits; system theory; discrete, IC, and VLSI circuit design; multidimensional circuits and systems; large-scale systems and power networks; nonlinear circuits and systems, wavelets, filter banks, and applications; neural networks; and signal processing. Content will also cover the areas represented by the Society technical committees: analog signal processing, cellular neural networks, and array computing, circuits and systems for communications, computer-aided network design, digital signal processing, multimedia systems and applications, neural systems and applications, nonlinear circuits and systems, power systems and power electronics and circuits, sensors and micromachining, visual signal processing and communication, and VLSI systems and applications. Lastly, the magazine will cover the interests represented by the widespread conference activity of the IEEE Circuits and Systems Society. In addition to the technical articles, which may be seen as the centerpiece of the start-up plan, we plan also to cover Society administrative activities, as for instance the meetings of the Board of Governors, Society People, as for instance the stories of award winners-fellows, medalists, and so forth, and places reached by the Society, including readable reports from the Society's conferences around the world.



Certified Chain of Custody
At Least 10% Certified Forest Content
www.sfpprogram.org
SFI-01681

Recognitions



Leon Chua Honored as Celebrated Member by IEEE Electron Devices Society

To honor and recognize esteemed Electron Devices Society (EDS) alumni, the EDS Celebrated Member Program has been instituted by EDS, and in 2019 this prestigious honor was bestowed upon Prof. Leon Chua of UC Berkeley, a Life CAS Fellow, for his enormous contributions to advancement of electron devices research. Only one member of the EDS is bestowed this honor per year and past Celebrated Members include luminaries such as Nobel Laureates George Smith, Herb Kroemer, and Leo Esaki, as well as Bob Dennard, Simon Sze, and Gordon Moore.

Leon Chua is widely known for his contributions to nonlinear circuit theory and the invention of the Memristor, Cellular Neural Networks and the Chua's Circuit. His research has been recognized internationally through numerous major awards, including 17 honorary doctorates from major universities in Europe and Japan, and 7 USA patents. He was elected as Fellow of IEEE in 1974, a foreign member of the European Academy of Sciences (Academia Europea) in 1997, a foreign member of the Hungarian Academy of Sciences in 2007, and an honorary fellow of the Institute of Advanced Study at the Technical University of Munich, Germany in 2012. He was honored with many major prizes, including the Frederick Emmons Award in 1974, the IEEE Neural Networks Pioneer Award in 2000, the first IEEE Gustav Kirchhoff Award in 2005, the International Francqui Chair (Belgium) in 2006, the Guggenheim Fellow award in 2010, Leverhulme Professor Award (United Kingdom) during 2010–2011, and the EU Marie Curie Fellow award, 2013. Prof. Chua is widely cited for the 12 hugely popular lectures he presented at the HP Chua Lecture Series, entitled "From Memristors and Cellular Nonlinear Networks to the Edge of Chaos," during the fall of 2015, and now accessible through YouTube via the link: <https://youtu.be/Rg3Vsc515YY>

CAS Society Awards 2019

The CAS Society is pleased to announce the recipients of the 2019 CAS Awards as follows:

Mac Van Valkenburg Award

Prof. Ángel Rodríguez-Vázquez

Charles A. Desoer Technical Achievement Award

Vivek K. De

John Choma Education Award

John Maxwell Cohn

Vitold Belevitch Award

Not awarded

Industrial Pioneer Award

Ulrich L. Rohde

Meritorious Service Award

David Allstot

Guillemin-Cauer Best Paper Award

Not awarded

Darlington Award

Not awarded

BioCAS Transactions Best Paper Award

Fabio Pareschi, Pierluigi Albertini, Giovanni Frattini, Mauro Mangia, Riccardo Rovatti, and Gianluca Setti, "Hardware-Algorithms Co-Design and Implementation of an Analog-to-Information Converter for Biosignals Based on Compressed Sensing," *IEEE Transactions on Biomedical Circuits and Systems*, Volume 10, Issue 1, Feb. 2016.

CAS for Video Technologies Transactions Best Paper Award

Han Hu, Yonggang Wen, Tat-Seng Chua, Jian Huang, Wenwu Zhu, Xuelong Li, "Joint Content Replication and Request Routing for Social Video Distribution Over Cloud CDN: A Community Clustering Method," *IEEE Transactions*

CAS Society News

By Gabriele Manganaro, Mohamad Sawan, and Yong Lian

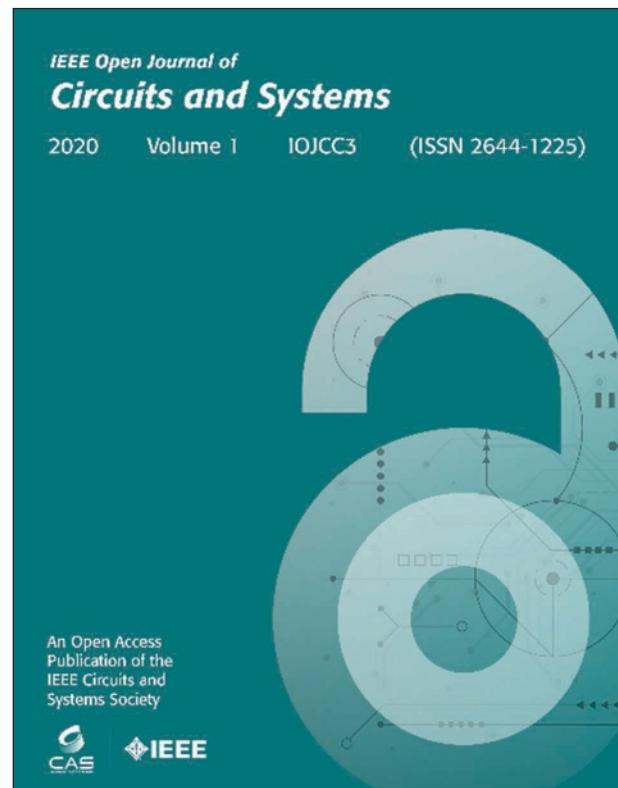
Launching the Open Journal of Circuits and Systems

In June, the IEEE announced the launch of 14 Open Access (OA) Journals, hence adding onto the other OA journals that some of the other IEEE societies have been successfully offering over the past few years. The IEEE Circuits and Systems Society (CASS) is among these 14. CASS has just started up one of these journals, called “The IEEE Open Journal of Circuits and Systems (OJ-CAS).” This is a new gold fully open access journal, spanning the full scope of the IEEE CASS’s field of interest, and that, as in the other cases, will only publish online articles.

OJ-CAS is ready to accept manuscript submissions and prospective authors will find relevant information and instructions on the CASS website, or simply going directly to the OJ-CAS webpage: <http://www.ieee-cas.org/publications/open-journal-circuits-and-systems>. Articles will be published in the form of regular papers and short papers, in 10–12 pages and 5–6 pages in length in the two-column IEEE Transactions format respectively.

To be considered for publication, all submitted manuscripts will be peer-reviewed with the very same procedures, scientific and ethical rigor of our existing CASS transactions. Such standards of scientific quality will be upheld by OJ-CAS’s Editorial Board, composed of a truly outstanding team of accomplished experts in CAS fields, most of them have previously served in the same or analogous role, and all of them are equally committed to living up to the same output and service to readers that distinguishes our journals. We are grateful for their volunteering service and personal commitment. The list of members of Editorial Board can be found at the above referred OJ-CAS webpage.

Just like all the other IEEE Open Access journals and being, indeed, accessible without restrictions, OJ-CAS will publish on IEEE Xplore, protected under the Creative Commons Attribution License (CC-BY), without requiring IEEE membership from readers or journal subscription fees. That is in contrast with most of the



traditional IEEE journals, requiring readers for a subscription or a direct payment per paper to gain access to the articles.

Much has been written about the differences between OA journals and subscription publications and we defer the reader to much more in-depth coverage of this matter, widely available on multiple sources. But a few important points deserve attention.

First off, OJ-CAS will be fully compliant with funder mandates, including Plan S, an international consortium of research funders, requiring, from 2021, that scientific publications resulting from research funded by public grants must be published in compliant Open Access journals or platforms. Similar initiatives are ongoing or in the making and, as such, CASS owes it to our community to be in front of that and OJ-CAS allows authors to meet this critical requirement.

Additionally, and no less important, the unrestricted access of articles published on OJ-CAS will mean that the reader audience will be intrinsically much broader than any of the subscription-based journals. It is not surprising that open access papers inherently tend to have a greater measured scientific impact than those that are not, everything else being equal.

Along these lines, some interesting consideration should be made. The CAS Society distinguishes itself from some other IEEE societies for an important aspect. That is the strong multidisciplinary nature of many of the CAS topics. The straightforward accessibility of OJ-CAS papers, also to non-IEEE members, goes hand in hand with CASS's multidisciplinarity. As an example, think of the mutual impact to advances in biotechnologies, or in neurosciences, when non-IEEE researchers in academia or industry will be able to gain access to what our Bio-medical Circuits and Systems (BioCAS) community, or our neuro-morphic technologists, will publish next on OJ-CAS. Or think of the higher cross-pollination that can be ignited between our published research in nonlinear circuits and systems on OJ-CAS and non-IEEE journals such as *Nature*, or *Acta Mathematica*, if only the barriers due to subscriptions are lowered. Please note that this is not a blue-sky aspiration. Academic libraries and corporate subscriptions have finite budgets everywhere and will tend to spend it on fewer specialized subscriptions, leaving all other readers to fetch high-quality research publications on open access journals. This is the present and it is poised to extend.

The cost aspect brings us to the last important point. Namely that, unlike traditional subscription-based journals, the costs of publishing on an OA journal, including OJ-CAS, fall on the authors. Provided that, after peer-

review, a manuscript has been accepted for publication, the authors are nominally responsible for the payment of an article-processing charge (APC) and that is not negligible (please visit OJ-CAS's website). However, APCs are often financed by an author's institution or the funder supporting their research. Which means that, depending on the individual circumstances, the net costs to the authors, after the institution/funder's contributions are accounted, will vary from similar (possibly less) dollar amounts as to a regular transaction paper, too much more if funding is unavailable. The financial agencies are in process to change the granting budgets to include the due APCs. Also, to come forward to you, discounts to authors that are CAS members are offered. It should also be noted that a paper's APC is only one among the many other components of a research budget. As engineers, it is in our nature to evaluate trade-offs and determine where to invest or to pivot for the max desirable outcome. With all that said, prospective authors should carefully evaluate their specific situation before embarking into an article submission, avoiding rather problematic circumstances with insufficient funding, should the peer-review result into a positive outcome. After all, an open-access published paper has a higher potential to bring more benefit (readers and citations) to authors than regular published same paper.

In conclusion, we are truly excited about the introduction of the IEEE Open Journal of Circuits and Systems. This puts CASS in a very strong position with the rapid proliferation of open access peer-reviewed publications, serving our members, opening the doors to a far broader audience and facilitating further knowledge synergy by lowering some access barriers.

IMAGE LICENSED BY GRAPHIC STOCK

We want to hear from you!

Do you like what you're reading?
Your feedback is important.
Let us know—send the editor-in-chief an e-mail!

IEEE



From Behavioral Design of Memristive Circuits and Systems to Physical Implementations

©EN.WIKIPEDIA.ORG/MEMRISTOR

Nima TaheriNejad and David Radakovits

Abstract

Since Hewlett Packard (HP) announced the passive fabrication of their memristors, various memristive technologies—as a promising emerging technology—have gained ever-increasing attention from the researchers. Although a natural application is using them as memory units, there have been several works in the literature showing their utilization in circuits and systems. While research on various aspects of memristive circuits and systems has been proliferating, the majority of these works are

based on simulations at different levels of modeling abstraction. Simulation is a very helpful design tool, and there have been several efforts in modeling memristors; however, we contend that at this point these simulations represent the reality of the behavior of memristors, especially in a circuit or system set-up, only to a very limited extent. We show how this negatively affects the reproduction of designed circuits and systems in different simulation levels, and more importantly in a real-world set-up with physical implementation. Following that, we look into some considerations which can improve the reproducibility of the circuits and systems to be designed in the future. We conclude the paper by suggesting certain approaches to tackle these practical challenges at device level as well as circuit and system level.

Digital Object Identifier 10.1109/MCAS.2019.2945209
Date of current version: 19 November 2019

I. Introduction

Reducing energy consumption is a crucial goal in the current circumstances of rapidly growing computational load. Mobile systems such as smartphones, embedded systems, wearable electronics, and Internet of Things (IoT) devices, which are often powered by batteries or rely on energy harvesting, require optimal utilization of the available energy [1]. A key factor in the energy budget of modern computing devices is memory [2]–[4]. In a modern chip, the number of transistors required to store data has a significant and increasing impact on the total transistor count [5], and consequently on the production cost. A promising solution for these problems is using memristors.

Although memristive behavior has been observed before [6], [7], a turning point for this type of basic circuit elements was when Hewlett Packard (HP) presented some of the (circuit level) applications of their passive solid-state Resistive Random Access Memory (ReRAM) devices with memristive characteristics in 2008 [8]. They advocated the memristors and their applications in the scientific community, especially that of circuits and systems. Thanks to their non-volatility, memristors could decrease the overall power consumption of the system dramatically [9]. Moreover, the relatively simple structure of memristors, allows compact implementations (device size of sub $10\text{ nm} \times 10\text{ nm}$ [10] and $3\text{ nm} \times 3\text{ nm}$ has been already reported [11], [12]) which can reduce their size up to one tenth of their Random Access Memory (RAM) counterparts [13]. It is worth noting that memristive behavior is not limited to ReRAMs. Other devices such as Phase Change Memory (PCM) and Spin Transfer Torque (STT) also show similar behaviors as described by Leon Chua in 1971 [14]. Since in many cases these devices face similar challenges, in this paper, we refer to them under the umbrella term of “memristors.” However, one should keep in mind that each of them has a different mechanism of operation which needs to be taken into account while working with them.

A natural candidate application for memristors has been in memory systems [9], [13], [15]–[22]. The possibility of integrating 1 TB of storage on a single chip [23] makes this technology a very attractive candidate for memory-intensive big-data applications [24]. Especially given that they can be integrated with Complementary Metal-Oxide Semiconductor (CMOS) technology with minimum changes. For examples, see MOSIS C5 CMOS [25], or CMOS Back End Of Line (BEOL) Memristor service [26], or the news on the planned offering of Taiwan

Semiconductor Manufacturing Company (TSMC) in 2019 [27]. More importantly, they can be used for purposes other than secondary memory, code-storage, or similar conventional ways non-volatile memories are often used [24]. There have been several efforts in using memristors for implementing various logical functions [28]–[44], calculations [38], [45]–[49], and other applications [50], [51] such as learning [51]–[55] and even cancer detection [56]. Fig. 1 summarizes some of the major events in the memristive community since 2008 [57].

Logic circuit design is the key to the development of memristor-based computing systems. A noteworthy observation in this regard is that the majority of these logics are inspired by CMOS and in a certain way mimic behaviors of a CMOS circuit or replace parts of it. Among these logics CMOS/Memristor Threshold Logic (based on the Logic Threshold Gate (LTG)) [28], [37], Ratioed Logic [28], [30], and CMOS-like Logic [28] are among the most prominent ones. One of the major common properties of all of these logics is that they operate in the voltage domain (information and logical states are represented in the voltage of a node). This includes some more recent ones like Scouting Logic [44] too. Due to the voltage representation of the values, they often compete with their traditional and far more mature CMOS counterparts. However, memristive technology can be more successful if its native properties such as having memory are exploited. This idea has been used in IMPLY Logic [31], [35], [64] and Memristor-Aided Logic (MAGIC) [39], [43] which fundamentally operate in the memory domain. That is, the information and logical

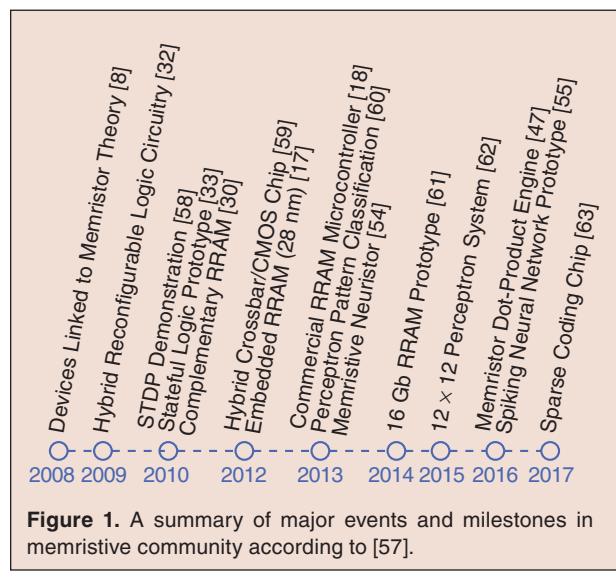


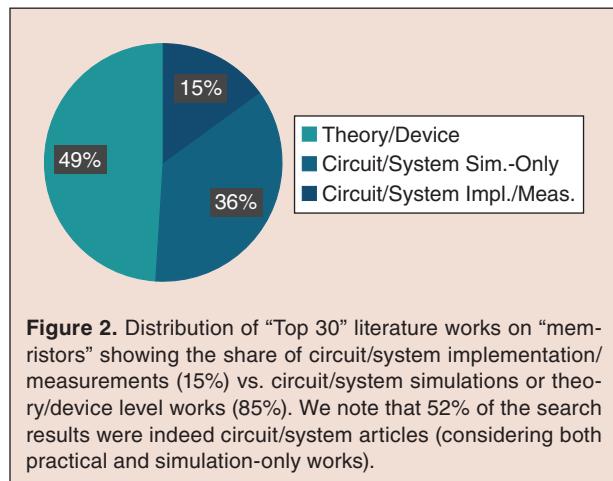
Figure 1. A summary of major events and milestones in memristive community according to [57].

Nima TaheriNejad and David Radakovits are with the Institute of Computer Technology, TU Wien, 1040 Vienna, Austria (email: nima.taherinejad@tuwien.ac.at, david.radakovits@tuwien.ac.at).

Among the top articles in the memristive circuit and system literature, less than 15% are verified by implementation and measurement.

states are represented as the memory states stored on the memristor. Therefore, no memory read or write is necessary before and after the logical operations. This is a substantial change compared to how CMOS circuits operate. Operating in memory domain is where CMOS has a significant disadvantage with respect to memristive technologies, since it requires a significantly larger area as well as power consumption to do a similar operation. Consequently, in-memory operation is the most propitious path for memristors as an emerging technology, especially considering the von-Neumann bottleneck which regards the constraints in data transmission between memory and processing units [57]. Few recent studies [42], [65]–[69] performed on in-memory computations have already shown promise.

The rest of this paper is organized as follows; In Section II, we first present our findings regarding the proportion (or from our point of view, in this case, disproportion) of physical implementation and simulation among memristor-based systems and circuits in the literature. Then, in Section III, we argue as to why this disproportion is very important in this field and should be ameliorated in the future. Next, in Section IV, we delve deeper into the shortcomings of models and some designs by presenting three examples. These experiments show how proper operation of memristive circuits cannot be taken for granted as it is often done in many memristive system and circuit designs. In Section V, we provide some suggestions as potential directions to take in order to tackle existing challenges. Finally, we draw our conclusions in Section VI.



II. Physical Implementation & Verification in the Literature

Studying the literature, to our surprise, there is a considerable shortage of physical implementation and verification among the myriads of circuits and systems proposed for memristive devices. To understand the dimensions of this shortage better, we conducted a survey in the literature. In this survey we searched five keywords, namely ‘memristor’, ‘memristor circuit’, ‘memristor system’, ‘memristor device’, ‘memristive circuit’, and ‘memristive system’, and picked the first thirty results (considering papers appearing in more than one search only once and dismissing it in subsequent appearances) and checked whether they were verified based on physical implementation or not. To make sure that our sample is not biased or limited to a certain community we chose “Google Scholar” as our search engine. “Google Scholar” searches a wide range of available materials including world-wide patents databases, research databases (such as IEEE, Elsevier, Springer, ResearchGate, ArXive, and others), university databases (for theses), and other published papers on the web (white papers published by companies or papers published by the authors on their personal website). Our results, visualized in Fig. 2, show that from the 142 works (38 of the 180 results were repetitive appearances), 51% were indeed circuit and system designs, however, only 30% of them (15.5% of all the search results) were based on physical implementation or measurements.

Knowing that some researchers prefer to use specific device names, we ran the same experiments with new keywords. Given the better reception of ReRAM among various types of memristive device, we chose the following keywords; ‘RRAM’, ‘ReRAM’, ‘resistive RAM’, ‘resistive memory’, ‘resistance switch’, ‘resistance switching’, ‘resistive switch’, ‘resistive switching’. The result for this set of search shows that the majority of the papers using this keyword are at a device and model level, and not circuit or system design. From the 211 works, only 11% were circuit and system designs, of which 78% were physically implemented. This means that only 9% of the total number of works found using those additional keywords were implementations.

We also did a brief search with PCM, where out of the 30 results, 9 papers (30%) were circuit or system designs, 89% of those containing actual implementations, yielding 27% implementations in total. Surprisingly, none

**Memristive technologies are young and on their way to maturity.
Hence, the path to maturity is no less long for the
models describing them.**

of these top 30 papers are concerned with modeling, even though the majority of them are device level works.

Considering the overall search combining the findings using all of above keywords, visualized in Fig. 3, the share of circuits and systems verified by physical implementation and measurements is as low as 12% (48 out of a total of 383 works)¹. We contend that this disproportion between practical works and simulated designs is alarming and has negative effects. Our search shows some indications that in communities using PCM and ReRAM keywords, they are more aware of (and hence pay more attention to) the importance of practical implementation. However, the ratio of circuit and system works done in those communities seems to be lower.

We note that such disproportion between simulation and implementation exists in certain other communities such as digital CMOS circuit design too. However, it is important to note that in that case, such disproportion is justified by the maturity of the models and Computer-Aided Design (CAD) tools developed for CMOS technologies thanks to the heavy investment of major companies and consequently comprehensive and continuous effort of engineers and scientists in device and circuit development and modeling. Therefore, a successful simulation is, to a large extent, a good guarantee of a successful implementation for the majority of digital CMOS circuits. Despite the maturity of CMOS technology and respective models and CAD tools, a successful simulation is much less of a guarantee for a successful implementation in the case of analog CMOS circuits. Therefore, in the respective community, new designs, and concepts are often well-received and commonly spread, if and only if they are backed up by physical implementation and verification. In consequence, the proportion of high-quality publications verified by physical implementation is much higher compared to their digital counterpart.

III. The Pitfalls of Circuit & System Simulation

We contend, that in the memristor community the shortage of physical implementation and verification is an important problem due to three main reasons, which we discuss here.

¹The share of circuit and system simulations-only is 15% (57 out of 383 non-repetitive works).

A. Immaturity of Technology

In the case of CMOS transistors, the material and fabrication process is to a very large extent determined and clear. The exact technology and fabricating company determine further details. When it comes to memristors, in contrast, this can vary significantly in terms of material, fabrication, and operation dynamics. Some of the circuits and systems are designed for “memristors” without any considerations for specifics of operations of different device types. Even among a single type, for example, ReRAMs, although Titanium dioxide (TiO_2) is one of the more well-known types, the literature contains many other materials and fabrication processes such as Tantalum Oxide (TaO_x) [70], [71], Hafnium Oxide (HfO_x) [72]–[74], amorphous silicon [75], carbon nanotubes [76], ferroelectric [77], SiNW [56], and silver-based ReRAMs [78], [79]. Given the sparse and ad-hoc approaches towards developing these memristors, they often remain in a pre-mature or maturing phase. Significantly less mature compared to any CMOS technology of the day. This negatively affects their characterization as well as their reproducibility. Some of the main challenges to be addressed at device level include device variability, cyclic variability, OFF/ON ratio, endurance, retention, and device speed.

Fig. 4 shows our measurement results performed on 8 ReRAMs of the same technology, fabrication round, and die packed in a single package². In our measurements,

²These memristors use metalization of chalcogenide material as the switching mechanism. We do not have permission to publish more specific details.

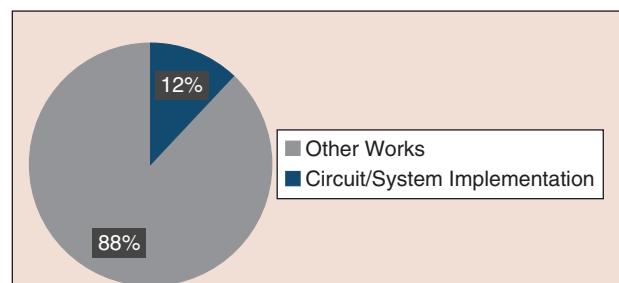


Figure 3. Distribution of “Top 30” literature works with all keywords (derived from memristor, ReRAM, and PCM) showing the share of circuit/system implementation/measurements (13%) from all articles found in our searches. Other works include theoretical papers, device level, or circuit and system designs which were not implemented (measured).

20 pulses of 800 mV amplitude and 25 μ s width were used to drive the memristors to their ON state (or low resistance, shown in solid circles) and 20 pulses of -400 mV amplitude and 25 μ s width were used to drive them to the OFF state (or high resistance, denoted by hollow circles). Each memristor was driven to the two opposite states and measured ten times for each state. We see that for the first 3–4 memristors (each column/group depicts the measurements of one memristor), shown in (shades of) blue, R_{on} and R_{off} values overlap in some measurements. This trend improves with a somewhat smaller or larger gap between the two values for other memristors, shown in different shades of green. Using memristors with overlapping R_{on} and R_{off} values is practically impossible. Therefore, either they should be identified beforehand and avoided, or other mechanisms should be devised to drive them into two distinct states. We note that the ratio of R_{off}/R_{on} also varies from 57 to 1. This wide range of variation makes working even

with “functional” memristors (that is, memristors without overlapping values of R_{on} and R_{off}) very difficult. Another difficulty is the variation in absolute values. The minimum R_{on} in this set of measurements is 17 k Ω , whereas the maximum was 1.17 M Ω , which shows a $\times 68$ difference. The smallest observed R_{off} was 172 k Ω and the largest 1.59 M Ω , a $\times 9$ difference, which is very large but significantly better than variations in R_{on} . It goes without saying that as the technology matures, we can hope to see more of the green memristors than the blue ones and more uniformity in fabrication.

B. Immaturity of Models

As mentioned, referring to memristors does not imply any specific technology, material or fabrication process. Even more specific terms such as ReRAM (or PCM) do not imply the same material or properties in that category. Various memristors have different properties and since they are often fabricated in research laboratories in small quantities, they are not available to other researchers and potential users for additional tests, experiments, or modeling. Consequently, the quality, verifiability, and scalability of respective models remain often very limited due to restricted experiments and modeling efforts possible at the research institute fabricating the memristor. A major issue, in this case, is that most models are developed at single device level under a few typical test scenarios (such as characterizing hysteresis loops, e.g. [80]–[83]). Although these models capture certain characteristics of the memristors, they rarely manage to sufficiently predict the behavior of the device under real application scenarios where the usage is substantially different from the tests. Moreover, the interaction between the devices and the environment is often neglected, leading to further inaccuracies when it comes to circuits involving more memristive devices, working in an uncontrolled environment.

We should bear in mind that even if the basic principles of operations of the memristor are known to us, some physical details, especially regarding the switching process, are yet rather unknown [6]. Based on this premise, very recently Menzel et al. [6] conducted an investigation of the quality of various prominent models on modeling the generic behavior of Redox-based memristive devices. This behavior includes voltage and current characteristics (I-V/I-t), non-linear switching kinetics, complementary resistive switching, multi-bit data storage, state-dependency, fading memory capability (asymptotic behavior), and model flexibility. Table I provides a summary of their study. We note that the performance is not measured against any real memristor, rather against generic behaviors of Redox based memristors (thus excludes any other types). Therefore,

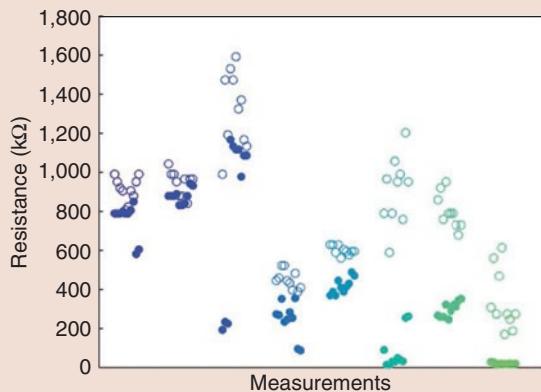


Figure 4. R_{on} and R_{off} measurement results for 8 ReRAMs in a single package. Each column represent measurements of one memristive device. Solid circles denote ON resistance, whereas hollow circles mark OFF resistances.

Table I.
Comparison of generic behaviors of four prominent models [6]. Where ○, ◉, and ● show unacceptable, acceptable, and good performance, respectively.

| | I-V/I-t | Kinetics | CRS | Multi-Bit | State-Dep. | Fading | Flexibility |
|---------------|---------|----------|-----|-----------|------------|--------|-------------|
| Strachan [87] | ● | ● | ● | ● | ● | ● | ○ |
| VTEAM [88] | ○ | ○ | ● | ○ | ○ | ○ | ● |
| Stanford [89] | ● | ○ | ● | ● | ● | ● | ● |
| RWTH [90] | ● | ● | ● | ● | ● | ● | ● |

**Be aware of limitations! Abstractions for high-level simulations
are very little reliable, so are the low-level simulations
using only nominal values.**

it does not show how good or bad they may model a specific physical memristor (of this or any other type). Additionally, this set of criteria does not provide a full insight to the behavior of a memristor.

As mentioned above, some models do consider the non-linearity and kinetics, which to some extent reflects the speed of the device. However, even for speed, more works such as extracting layout and wire parasitics and considering them in the models are missing. In addition, noise, variability, and temperature are also not reflected properly in the models. It is known that temperature can affect properties such as the mobility in the devices [57], which can, in turn, affect its behavior. This effect is stronger in some materials such as VO_2 and NbO_2 , leading to the observation of negative differential resistance [57]. A property which has been used to create oscillation [84] and neurons [54], [84]. Although this behavior for these specific types has been modeled [85], [86], main-stream generic models do not consider any temperature effects.

C. The Nature of Designed Circuits

Memristive circuits such as neuromorphic systems [24], [53], [60], [91] or Threshold Logic [28], [37], and some other custom circuits [53], [92] treat memristors as a device with a continuous range of values. This is in nature similar to analog circuits rather than digital circuits and therefore, requires similar care and approach as in analog circuits. That is, new designs and concepts need to be backed up by physical implementation and verification in order to be reliable. On the other hand, in many of the digital memristive circuits, a highly non-linear behavior (sharp switching based on thresholds) is assumed and used. In reality the extent of this non-linearity is more limited and the analog continuous nature of memristors has a strong presence, making a notable difference in practice.

For example, in IMPLY Logic [31], [35], [64], the state change operation heavily depends on the voltage difference between V_{COND} and V_{SET} , which is significantly smaller than the power supply, providing a very weak non-binary drive for the change. On the other hand, it is assumed that if the voltage across a memristor is below the threshold, that memristor will not experience any changes. Whereas, in practice, that memristor can experience a state drift (we will show and discuss this more in Section IV-B). Although the input and output are considered as digital values, the dynamic of the operation is significantly closer to the traditional analog circuits

than the digital circuits. Again, requiring considerations that are common in the analog domain but are often omitted in the digital domain.

IV. Reproducibility Challenge

Shortcomings discussed in Section II and Section III negatively affect reproducibility of designs in the real world and consequently, put a question mark on the extent of their practicality and usefulness. In some cases, these issues lead to inconsistencies and reproduction issues in simulations as well, which poses an even greater challenge to the community. We have tried to reproduce some of the existing works in the literature and faced certain problems which we briefly report here. We note that to answer questions such as “which logic is more reliable in practice?” a comprehensive set of studies is in order. In such a study one should test them in practice at the presence of practical challenges, and assess the difficulty to come up with solutions to resolve any problems they may face in practice. Only then a good perspective of the advantages or disadvantages of one logic or another can be discussed. Even though, we hope that this paper and particularly this section will ignite such discussions and plant the seed for such studies, such in-depth analysis and discussion is not in the scope of the present paper. Here we narrate our limited experiences to highlight the challenges of practical implementation and attract the attention of researchers and engineers to their importance while leaving comprehensive comparative studies for future works.

A. Memory Example

There are several papers in the literature on various aspects of using memristors as memories [9], [13], [15], [16], [19]–[22]. Although various models capture different characteristics of memristors, interestingly enough, so far and to the best of our knowledge, none of the existing ones model the leakage or state drift fully and properly. A factor that substantially affects the performance of memristors as memory. We bear in mind that there have been works on “history effect” [93], [94], however, that is a different concept. History effect concerns the eventualty of a steady state in memristors, which is independent of its initial state, after application of a certain input pattern. In studying history effect, the authors explicitly mention that they do not consider state changes in the absence of input [93], whereas the leakage or

Real devices show behaviors that may not be represented by current models, yet they can affect the function of the circuit and the system.

state drift discussed here regards the state changes in the absence of any input.

There are several suggestions regarding compensation of read-out mechanism, nonetheless, there are no concrete solutions addressing the leakage. To evaluate these effects we designed and implemented (shown in Fig. 5) a memory write and read circuit [95] and measured the maximum retention time of Knownm “BS-AF-W” memristors [96]. Our measurement showed 81 hours of retention time (488 reads, 10 minutes apart) for our memory system. In our experiments, the read operations were not compensated and hence could affect the retention time. However, in a different experiment with more frequent reads (every 1 second), we managed to have 6000 correct reads before a state change. Therefore, even though the 488 read operations contribute in reduction of retention time, the two experiments prove the presence of a different effect which we associate with the leakage. Since currently, no models for this effect exist, we could not simulate this effect. Similarly, no other memristive memory system design can be thoroughly verified in simulation (e.g., regarding its retention time). At least not using the models currently available to the public and academia. This indicates the need for developing new and more comprehensive models. More importantly, it highlights the importance of the physical implementation and verification of memristive systems. We note that there are memory products in

the market, e.g., [97], which use memristors. Therefore, there is no doubt that there are and can be more memristive memory systems, for which similar characterization experiments are done. However, we could not find any similar reports accessible to the public, reporting those important characteristics we measured.

B. Logic Example—IMPLY

As one of the most prominent memristor-based logic design methods, IMPLY has been extensively used [31], [34], [38], [48], [49], [99]–[102]. There are also various works in the literature on its design and implementation as well, e.g. [34], [35], [38], [45]. This also includes closed-form formulas and determined boundaries regarding the value of various circuit elements necessary to implement an IMPLY [38], [45]. In the literature, however, often the circuit simulation of systems using IMPLY is skipped (see e.g., [45], [48], [99], [100]), given the assumption that the basic gate implementation as shown in [34] is functional. However, to verify our IMPLY based system published in [48], we tried to run the respective simulation in SPICE. The common assumption in $p \rightarrow q$ (p IMPLY q) is that p maintains its state while q changes its state to hold the result [31], [45], [48], [100]–[102]. Despite the fact that many of the designed systems work strictly based on this assumption, to our surprise, this does not seem to be always the case.

One of the problems with the closed-form calculations is that they assume a resistive switch with fixed resistance before and after crossing the threshold voltage. Whereas in reality, the memristors experience state drift on both sides of this threshold, which on itself affects the switching process. According to our simulations using four different models (namely, Biolek [98], Yakopcic [103], Joglekar [104] and TEAM [105]), in a single operation, or as shown in [34] in a few step operation, the parameters could be set such that p can be considered (in some cases only marginally) as keeping its previous states. However, the state drift in a sequence of operations leads to a potential loss of state for p and consequently false results (see Fig. 6). The only model in which we managed to simulate IMPLY with a small enough state drift that does not cause a loss of state is TEAM [105]. However, we notice that for doing so we had to set the model parameters arbitrarily and far from the characteristics of the real memristors we have at hand.

Taking all the above into account, the question of the practicality of a physical implementation of IMPLY



Figure 5. Our implementation of a complete memory system on PCB, populated with memristor chip and required circuits to read and write.

in which p does not lose its state after a few operations remains an open question to us. However, this problem can be easily remedied. That is, to save the content of the p , whenever it is needed in the future. Despite the overhead, this approach seems to be practically more attainable.

C. Logic Example—MAGIC

Another prominent memristor-based logic is MAGIC [39], [43]. Several logic functions can be implemented using MAGIC, one of them being an n -input NOR, which naturally forms a NOT gate if only one input is used. Given its attractiveness, we have tried to implement it. This logic also faces several challenges in overcoming practical adversities of memristive circuit implementations.

Some of these challenges can be observed already in simulations. For example, our simulations of MAGIC NOR gates have shown that variation in memristor parameters, i.e., R_{on} , R_{off} , threshold current/voltage, and switching dynamics, cause the robustness to decrease dramatically. Simulations of these circuits were conducted using VTEAM [88]. Since VTEAM is available as a Verilog-A model only, we implemented it in LTspice. The implementation can be found in [106]. The used parameters for VTEAM can be found in Table II. Fig. 7 shows a sample result of our simulations, where the effect of the initial state on the performance can be seen, particularly timing. For example, as it can be seen in this figure, a 5% deviation from a 100% initial state leads to a double inversion time. Not taking this into account can lead to incomplete state changes and eventually false results in operations. It is appealing to think that by taking a very long operation time this problem should go away, how-

ever, we need to bear in mind that a longer operation time leads to a larger state drift in the input memristors. Therefore, even though the state change in the output memristor would be thus improved, a similar problem would be introduced to the input memristor which would bring us back to the same problem in further operations.

Moreover, as we showed in Section III-A, the mismatch between memristors, which can have a similar effect as to not fully ON or OFF initial states, could be more than an order of magnitude larger than what is shown here. Therefore, finding pairs or groups of memristor with similar enough properties to implement the gate is challenging too. Even harder is their inter-operation. Whereas we successfully implemented MAGIC NOT gates (i.e., gates involving two memristors), due to parameter variations existing in the memristors available to us, it was barely possible to implement 2-input NOR gates in MAGIC. It is important to notice that our implementation of this gate did not prove to be particularly reliable and repeatable.

We would like to remark, that this does not undermine the value and the promise of this design. The question for us (and we believe the rest of the community) is

Table II.
Parameter values used in VTEAM.

| Parameter | V_{off} | V_{on} | α_{off} | α_{on} | R_{off} | R_{on} |
|-----------|-----------|-----------|----------------|---------------|-----------|----------|
| Value | 0.7 V | -10 mV | 3 | 3 | 1 MΩ | 10 kΩ |
| k_{on} | k_{off} | w_{off} | w_{on} | w_c | a_{off} | a_{on} |
| -0.5 nm/s | 1 cm/s | 3 nm | 0 nm | 107 pm | 3 nm | 0 nm |

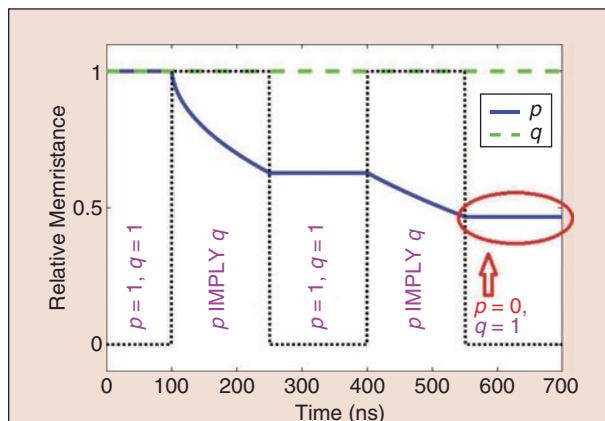


Figure 6. Simulation of two consecutive IMPLY operations without refreshing state of memristors (this example uses Bi-olek model [98]). Memristor p (blue solid line) loses its state during the operations (black dotted line going to “1” shows when IMPLY operation was performed), while q (green dashed line) keeps its state as expected.

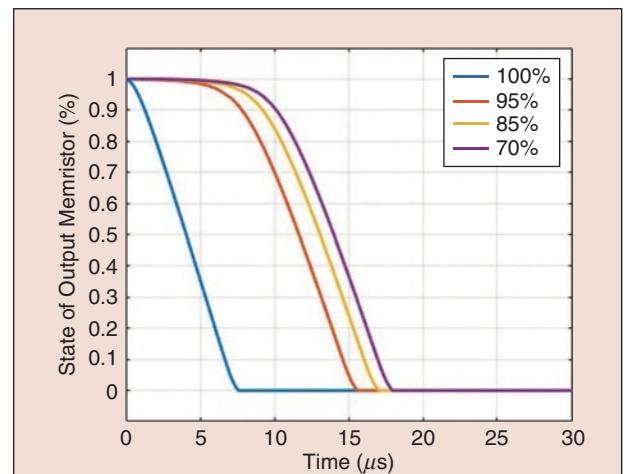


Figure 7. Dependability of the result of a MAGIC ‘NOT’ operation on the initial state of the input memristor. The NOT operation should yield $in = 1 \rightarrow out = 0$, but if the state of $in = 0.95$ it already consumes twice the time to reach $out = 0$, compared to the fully $in = 1$ state. After the original operation time of 8 μ s, the $in = 0.95$ yields $out = 0.89$ (which is a false result).

how a more reliable and repeatable implementation of MAGIC can be made possible? Be it a solution at device level (e.g., less variation could help), circuit level (e.g., could any extra circuit help?), or system level. For example, at system level, the state drifts of input memristors due to longer operation time (which improves the state change in output memristor) can be compensated by refreshing the state of the input memristors (as we suggested for IMPLY too). Similar to the IMPLY example above, practical issues may be therefore overcome with certain considerations which affect the system at higher (e.g., at algorithm) levels too.

V. Moving Forward

In previous sections, we discussed major challenges that memristive circuits and systems face in practice. We would like to emphasize that this is not meant to undermine the practicality of building memristive circuits and systems. It has been repeatedly proven that it is possible to build memristive circuits which work in practice, for example, see [107] and [108], among many others. Our intention here is to raise awareness about these challenges and by considering these issues, empower engineers and designers to design circuits and systems which have a shorter path to practical implementations. Other than taking the challenges of the current state into account, there are certain steps that we, as a community, could take to alleviate these adversities and reduce the existing challenges. In this section, we summarize some of these potential steps, which provide formidable research questions and challenges. Addressing them could have a positive effect on the design and implementation of memristive circuits and systems. Some of the challenges to be addressed at the device level are:

Device Variability As we saw in Section A, particularly the example of Fig. 4, the variation between devices, even within the same package, is so large that the R_{off} of a memristor can be smaller than the R_{on} of another in the same package. This makes it impossible to consider even an arbitrary range of these values, within which both memristors can be considered either ON or OFF. Therefore, the biggest hurdle of practical implementations is this extremely large variation between device characteristics. Reducing the size of memristors seems to be a key solution to this problem. Material research also could lead to improvements [109]–[111].

OFF/ON Ratio Even in a single memristor, it is important to have a minimum of $R_{\text{off}}/R_{\text{on}}$ so that the two different states can be distinguished. As evident in our example, Fig. 4, this ratio undergoes a large variation too. Values equal to or smaller than 1, which speak of practically indistinguishable states, are the major problem. In addition, certain applications require much larger ratios.

This issue, even though has a different effect in practice, solution-wise, follows the previous issue closely. Solutions which could alleviate the problem of device variability could help in improving the OFF/ON ratio as well.

Cyclic Variability Variations during the lifetime of the memristor is a well-known phenomenon as well [112]. This challenge might be significantly more difficult than the device and ratio variation to address at the device level. Considering it at circuit and system level may be a somewhat easier approach. However, in the presence of the previous two issues, this challenge has little priority. Moreover, in many cases, its pattern is hardly distinguishable. For example, we could not see a particular pattern of cyclic variations in our measurements, however, once this pattern is distinguished and modeled, circuit and system designers could better consider it in their designs.

Endurance Currently, the lifetime of memristors is not very high in all technologies. In some cases, it could be as low as 10,000 cycles [112], [113]. That is certainly a limitation which could affect the wide-spread use of them, particularly in applications such as in-memory logic and computations which come with frequent changes of state. Therefore, device level research is needed to improve this aspect too. However, this problem seems to be of a secondary priority compared to device variability and OFF/ON ratio.

Retention This problem mainly concerns memory applications. However, given that in future architectures under investigation (which try to use in-memory computation), memristors are going to act both as memory and computation unit, retention becomes an important aspect for them as well. At this stage, this is also a secondary concern, but very important for the wide-spread use of these devices in consumer electronics. Material and device research seems to hold the answer to this question.

Device Speed In the literature a large range (from sub-nanosecond to microseconds) of device speeds can be observed [96], [114], [115]. This affects the performance and power consumption of the systems using memristive circuits and systems. Improving the speed of state changes at device level can make memristive circuits and systems more competitive in the CMOS-dominated market and hence improve its reception by the industry and users.

Sneak Path is a well-known issue in the literature [116], [117] and there have been efforts in reducing the effect of sneak path [118], [119]. For example, in [118] the authors use a system with buffer amplifiers to reduce the number of memristors which would normally be affected by the sneak path. Thus, they alleviate the problem, even though it does not go completely away. In addition, the complexity of the system is increased

While real devices are hard to access, awareness of potential challenges and thorough low-level simulations are crucial for a reliable design.

and issues such as amplifier noise are introduced. A simple solution is adding a transistor to switch a memristor in or out of the crossbar. The main problem with this structure, also called 1T1M, is the additional transistor gate wire which makes the crossbar less compact compared to a cell structure that requires only a memristor. The required transistors can have an even larger effect in the total area of the crossbar. Device level solutions such as larger OFF/ON ratios also could alleviate the sneak path problem. In that case, 1T1M could change from a functional necessity to a luxurious addition for applications which require higher reliability and can accept the additional costs of 1T1M structures.

At the circuit and system level, the following steps can help the design and implementation process;

Models Memristor models still have a long way to completion. Modeling temperature effects, device variations (particularly the variations in the absolute value of R_{on} and R_{off}), threshold variation, cyclic variations, leakage (retention time), and endurance are some of the practical effects which, to the best of our knowledge, are not reflected in any of the existing models. Modeling these effects — especially in one integrated model — can enable more realistic simulations, particularly corner simulations.

Parasitics More often than not, memristor models are developed in a laboratory environment and are based on on-die measurements. They do not consider any of the parasitics which can be formed due to the wires and connections as well as the layout of the circuits. Creating and using models for these effects can help in the design process and lead to more realistic simulations which better represent implementations.

Functional Simulations Using better models which reflect the reality better, in terms of values and variations and include leakage, extracted parasitics, and unideal initial states (memristors that enter an operation without having reached their full ON or OFF states in previous operations), the circuits can be thoroughly tested to see whether they are functional under all those circumstances or not. If not, the range of functional operations, as well as more problematic issues, can be identified. The former allows selecting suitable applications or technologies and the latter helps in devising solutions to address the relevant functional issues.

Corner Simulation Evaluating a circuit in different corners, such as the ones mentioned above, help in predicting the chances of prototypes being functional,

or selection of the technology to fabricate the designed circuit, as well as the universality of the design. That is, how much of variation in those parameters the design can tolerate before showing functional problems? Consequently, this helps to find suitable technologies since technologies which have a variation within those bounds can be used to implement that design.

Design and Test Awareness Once designers are aware of the challenges of practical implementations, as we discussed in this paper, they can design their circuits such that they can overcome or better tolerate these adversities. Whether these solutions be at circuit level, or at system level (e.g., the one we proposed here for the practical problems of the IMPLY gate). It is also important to test the circuits and systems against them, both in simulations and in practice.

Integration Many of working implementations are based on Integrated Circuit (IC) solutions [107], [108]. That is, the memristors and the CMOS circuits are on the same die or in the same package. This seems to be a possible solution for more reliable implementations. This facility is hardly available to the public but it seems to have a considerable effect since most published practical implementations are ICs. With the announcement of TSMC [27] regarding their new fabrication rounds which include memristive devices, this could change and we hope to see more practical implementations.

VI. Conclusions

Our experiments above show critical issues to which the community needs to pay more attention to create more effective and realistic memristive circuits and systems. First, is the necessity of developing improved, more comprehensive and more realistic memristor models which represent the behavior of real-world memristors better. In particular, factors such as device, threshold and cyclic variations, as well as temperature effects, leakage (retention time) and endurance. This point was presented in the example of the memory system we have developed. There, we showed factors such as retention time or realistic refresh cycles cannot be simulated since parameters such as leakage or device variation are not fully or properly represented in the existing models. Completion of models presents a longer-term challenge since the memristive technology itself is evolving. This process of evolution requires renewing respective models to better represent the physical behavior of newer technologies.

Second is the necessity of running full functional and corner circuit-level simulations for designed systems. Current circuit models have room for improvement and they do not provide a comprehensive insight as to what may happen in the physical world. Nevertheless, they do provide an understanding of certain potential problems in the designed systems or points where a deeper investigation is necessary. An example of this point is the IMPLY operation where, as presented here, most of the existing models show that with realistic model configurations, it is most likely that consecutive operations lead to loss of information on the p memristor. However, many system designs do not consider this potential loss of information which can have a significant (and potentially negative) effect on the operation of these systems. Therefore, certain considerations (such as refreshing the value of p in this example) are in order, to ensure the proper operation of higher level systems using IMPLY.

Lastly, we contend that many of the current memristive circuits, such as IMPLY for example, although having digital inputs and outputs, operate in a manner that can be identified better with analog CMOS circuit operations, rather than their digital counterpart. Therefore, to reliably verify their operation, and to realistically characterize them, they need to be physically implemented and tested. A matter that we believe is relatively overlooked in the community and it deserves more attention. These practical implementations show us the way forward to improve both devices and memristive circuits and systems.



Nima TaheriNejad (S'08-M'15) received his B.Sc. and M.Sc. degrees in electrical and electronic engineering from the Babol Nushirvani University of Technology, Babol, Iran, and the Iran University of Science and Technology, Tehran, Iran, in 2007 and 2009, respectively; and the Ph.D. degree in electrical and computer engineering from The University of British Columbia, Vancouver, Canada, in 2015.

He is currently a “Universitätsassistent” at the TU Wien (formerly known also as Vienna University of Technology), Vienna, Austria, where his areas of work include self-awareness in resource-constrained cyber-physical systems, embedded systems, systems on chip, memristor-based circuit and systems, health-care, and robotics. He has published two books and more than forty peer reviewed articles. He has also served as a reviewer, editor, organizer, and chair for various journals, conferences, and workshops. Dr. TaheriNejad has received several awards and scholarships from universities and conferences he has attended.



David Radakovits received his B.Sc. degree in electrical engineering and information technology from TU Wien, Vienna, Austria, where he is currently a graduate student. His B.Sc. thesis was on “Binary storage on Memristors” and he continues to work on memristive circuits and systems for his M.Sc. degree in Embedded Systems.

His main research interests are memristive circuits and systems, embedded systems and systems-on-chip, and hardware security. He has two published papers and two papers under review at IEEE journals and conferences on different aspects of memristive circuits and systems.

References

- [1] R. Iyer and E. Ozer, “Visual IoT: Architectural challenges and opportunities; toward a self-learning and energy-neutral IoT,” *IEEE Micro*, vol. 36, no. 6, pp. 45–49, Nov. 2016.
- [2] A. Carroll and G. Heiser, “An analysis of power consumption in a smartphone,” in *Proc. USENIX Annu. Tech. Conf.*, vol. 14. Boston, MA, 2010, p. 21.
- [3] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, “Full-system power analysis and modeling for server environments,” in *Proc. IEEE Int. Symp. Computer Architecture*, 2006.
- [4] A. Mahesri and V. Vardhan, “Power consumption breakdown on a modern laptop,” in *Proc. Int. Workshop on Power-Aware Computer Systems*. Berlin, Germany: Springer-Verlag, 2004, pp. 165–180.
- [5] *International Technology Roadmap for Semiconductors – System Drivers*, 2011th ed., ITRS Technology Working Group, 2011.
- [6] S. Menzel, A. Siemon, A. Ascoli, and R. Tetzlaff, “Requirements and challenges for modelling redox-based memristive devices,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, May 2018, pp. 1–5.
- [7] W. Rainer, D. Regina, S. Georgi, and S. Kristof, “Redox-based resistive switching memories nanoionic mechanisms, prospects, and challenges,” *Adv. Mater.*, vol. 21, no. 2526, pp. 2632–2663, 2009. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.200900375>
- [8] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
- [9] B. Mohammad, D. Homouz, and H. Elgabra, “Robust hybrid memristor-CMOS memory: Modeling and design,” *IEEE Trans. VLSI Syst.*, vol. 21, no. 11, pp. 2069–2079, Nov. 2013.
- [10] B. Govoreanu et al., “10×10nm² Hf/HfO_xcrossbar resistive RAM with excellent performance, reliability and low-energy operation,” in *Proc. Int. Electron Devices Meeting*, Dec. 2011, pp. 31.6.1–31.6.4.
- [11] A. Sinha, M. S. Kulkarni, and C. Teuscher, “Evolving nanoscale associative memories with memristors,” in *Proc. 11th IEEE Int. Conf. Nanotechnology*, Aug. 2011, pp. 860–864.
- [12] R. S. Williams, “Finding the missing memristor,” 2010. Accessed on: Nov. 6, 2017. [Online]. Available: <https://www.youtube.com/watch?v=bKGhvKyjgL>
- [13] H. Kim, M. P. Sah, C. Yang, and L. O. Chua, “Memristor-based multi-level memory,” in *Proc. 12th Int. Workshop Cellular Nanoscale Networks and Their Applications (CNNA)*, Feb. 2010, pp. 1–6.
- [14] L. Chua, “Memristor – The missing circuit element,” *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.
- [15] D. Niu, Y. Chen, and Y. Xie, “Low-power dual-element memristor based memory design,” in *Proc. ACM/IEEE Int. Symp. Low-Power Electronics and Design (ISLPED)*, Aug. 2010, pp. 25–30.
- [16] Y. Ho, G. Huang, and P. Li, “Dynamical properties and design analysis for nonvolatile memristor memories,” *IEEE Trans. Circuits and Systems I, Regular Papers*, vol. 58, no. 4, pp. 724–736, Apr. 2011.
- [17] W. C. Shen et al., “High-K metal gate contact RRAM (CRRAM) in pure 28nm CMOS logic process,” in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, 2012, pp. 31–36.
- [18] G. Hilson, “Imec, Panasonic Push progress on ReRAM,” *eeTimes*, 2015. [Online]. Available: https://www.eetimes.com/document.asp?doc_id=1327307

- [19] V. Baghel and S. Akashe, "Low power memristor based 7T SRAM using MTMOS technique," in Proc. 5th Int. Conf. Advanced Computing Communication Technologies (ACCT), Feb. 2015, pp. 222–226.
- [20] M. Zangeneh and A. Joshi, "Design and optimization of nonvolatile multibit 1T1R resistive RAM," *IEEE Trans. VLSI Syst.*, vol. 22, no. 8, pp. 1815–1828, Aug. 2014.
- [21] N. Taherinejad, P. D. S. Manoj, and A. Jantsch, "Memristors' potential for multi-bit storage and pattern learning," in Proc. IEEE European Modelling Symp. (EMS), Oct. 2015, pp. 450–455.
- [22] N. Taherinejad, S. M. P. D., M. Rathmair, and A. Jantsch, "Fully digital write-in scheme for multi-bit memristive storage," in Proc. 13th Int. Conf. Electrical Engineering, Computing Science and Automatic Control (CCE), Sept. 2016, pp. 1–6.
- [23] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature Nanotechnol.*, vol. 8, pp. 13–24, 2013.
- [24] S. Hamdioui et al., "Memristor for computing: Myth or reality?" in Proc. Design, Automation Test in Europe Conf. Exhibition (DATE), Mar. 2017, pp. 722–731.
- [25] "Neuro-bit," Bio Inspired Technologies LLC, 2017. [Online]. Available: <http://www.bioinspired.net/home.html>
- [26] Knownm, "Knownm Inc.," 2017. [Online]. Available: <https://knowm.org>
- [27] P. Clarke, "TSMC to offer embedded ReRAM in 2019," *eeNews*, 2017. [Online]. Available: <http://www.eenewsanalog.com/news/report-tsmc-offer-embedded-reram-2019>
- [28] I. Vourkas and G. Sirakoulis, *Memristor-Based Nanoelectronic Computing Circuits and Architectures: Foreword by Leon Chua* (Emergence, Complexity and Computation). Berlin, Germany: Springer-Verlag, 2015.
- [29] S. Muroga, *Threshold Logic and Its Applications*. New York: Wiley, 1971.
- [30] E. Linn, R. Rosezin, C. Kügeler, and R. Waser, "Complementary resistive switches for passive nanocrossbar memories," *Nature Mater.*, vol. 9, no. 5, pp. 403, 2010.
- [31] E. Lehtonen and M. Laiho, "Stateful implication logic with memristors," in Proc. 2009 IEEE/ACM Int. Symp. Nanoscale Architectures. IEEE Computer Society, 2009, pp. 33–36.
- [32] J. Borghetti et al., "A hybrid nanomemristor/transistor logic circuit capable of self-programming," *Proc. Nat. Acad. Sci.*, vol. 106, no. 6, pp. 1699–1703, 2009.
- [33] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "Memristive switches enable stateful logic operations via material implications," *Nature*, vol. 464, pp. 873–876, 2010.
- [34] S. Kvatinsky, A. Kolodny, U. C. Weiser, and E. G. Friedman, "Memristor-based imply logic design procedure," in Proc. IEEE 29th Int. Conf. Computer Design (ICCD), 2011, pp. 142–147.
- [35] E. Linn, R. Rosezin, S. Tappertzhofen, R. Waser, et al. "Beyond von Neumann logic operations in passive crossbar arrays alongside memory operations," *Nanotechnology*, vol. 23, no. 30, pp. 305,205, 2012.
- [36] L. Gao, F. Alibart, and D. B. Strukov, "Programmable CMOS/memristor threshold logic," *IEEE Trans. Nanotechnol.*, vol. 12, no. 2, pp. 115–119, 2013.
- [37] S. Kvatinsky et al. "Memristor-based material implication (IMPLY) logic: Design principles and methodologies," *IEEE Trans. VLSI Syst.*, vol. 22, no. 10, pp. 2054–2066, 2014.
- [38] S. Kvatinsky et al., "MAGIC—Memristor-aided logic," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 61, no. 11, pp. 895–899, Nov. 2014.
- [39] G. Snider, "Computing with hysteretic resistor crossbars," *Appl. Phys. A*, vol. 80, no. 6, pp. 1165–1172, 2005.
- [40] A. Siemon, S. Menzel, R. Waser, and E. Linn, "A complementary resistive switch-based crossbar array adder," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 5, no. 1, pp. 64–74, 2015.
- [41] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie, "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories," in Proc. 53rd ACM/EDAC/IEEE Design Automation Conf. (DAC), June 2016, pp. 1–6.
- [42] N. Talati, S. Gupta, P. Mane, and S. Kvatinsky, "Logic design within memristive memories using memristor-aided logic (MAGIC)," *IEEE Trans. Nanotechnol.*, vol. 15, no. 4, pp. 635–650, July 2016.
- [43] L. Xie et al., "Scouting logic: A novel memristor-based logic design for resistive computing," in Proc. IEEE Computer Society Annu. Symp. VLSI (ISVLSI), July 2017, pp. 176–181.
- [44] M. Teimoory, A. Amirsoleimani, J. Shamsi, A. Ahmadi, S. Alirezaee, and M. Ahmadi, "Optimized implementation of memristor-based full adder by material implication logic," in Proc. 21st IEEE Int. Conf. Electronics, Circuits and Systems (ICECS), 2014, pp. 562–565.
- [45] X. Wang et al., "Memristor-based XOR gate for full adder," in Proc. 35th Chinese Control Conf. (CCC), 2016, pp. 5847–5851.
- [46] M. Hu et al., "Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication," in Proc. 53rd Annu. Design Automation Conf. ACM, 2016, p. 19.
- [47] S. G. Rohani and N. TaheriNejad, "An improved algorithm for imply logic based memristive full-adder," in Proc. IEEE 30th Canadian Conf. Electrical and Computer Engineering (CCECE), Apr. 2017, pp. 1–4.
- [48] N. TaheriNejad, T. Delaroche, D. Radakovits, and S. Mirabbasi, "A semi-serial topology for compact and fast IMPLY-based memristive full adders," in Proc. IEEE New Circuits and Systems Symp. (NewCAS), 2019, pp. 1–5.
- [49] P. Mazumder, S.-M. Kang, and R. Waser, "Memristors: Devices, models, and applications," *Proc. IEEE*, vol. 100, no. 6, pp. 1911–1919, 2012.
- [50] Y. Pershin and M. Di Ventra, "Neuromorphic, digital, and quantum computation with memory circuit elements," *Proc. IEEE*, vol. 100, no. 6, pp. 2071–2080, June 2012.
- [51] Y. V. Pershin, S. L. Fontaine, and M. D. Ventra, "Memristive model of amoeba learning," *Phys. Rev. E*, vol. 80, pp. 1–6, 2009.
- [52] A. Thomas, "Memristor-based neural networks," *J. Phys. D, Appl. Phys.*, vol. 46, no. 9, p. 093,001, 2013.
- [53] M. D. Pickett, G. Medeiros-Ribeiro, and R. S. Williams, "A scalable neuristor built with Mott memristors," *Nature Mater.*, vol. 12, no. 2, p. 114, 2013.
- [54] V. Milo et al., "Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent plasticity," in Proc. IEEE Int. Electron Devices Meeting (IEDM). IEEE, 2016, pp. 16–18.
- [55] A. Zaher, P. Hfliger, F. Puppo, G. D. Micheli, and S. Carrara, "Novel readout circuit for memristive biosensors in cancer detection," in IEEE Biomedical Circuits and Systems Conf. (BioCAS) Proc., Oct. 2014, pp. 448–451.
- [56] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," *Nature Electron.*, vol. 1, pp. 22–29, 2018.
- [57] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, 2010.
- [58] K.-H. Kim et al., "A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications," *Nano Lett.*, vol. 12, no. 1, pp. 389–395, Jan. 2012.
- [59] F. Alibart, E. Zamanidoost, and D. B. Strukov, "Pattern classification by memristive crossbar circuits using ex situ and in situ training," *Nature Commun.*, vol. 4, 2013.
- [60] R. Fackenthal et al., "19.7 A 16Gb ReRAM with 200MB/s write and 1GB/s read in 27nm technology," in Proc. IEEE Int. Solid-State Circuits Conf. Dig. Technical Papers (ISSCC), 2014, pp. 338–339.
- [61] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, p. 61, 2015.
- [62] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, and W. D. Lu, "Sparse coding with memristor networks," *Nature Nanotechnol.*, vol. 12, no. 8, p. 784, 2017.
- [63] Q. Chen, X. Wang, H. Wan, and R. Yang, "A logic circuit design for perfecting memristor-based material implication," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 2, pp. 279–284, 2017.
- [64] R. B. Hur and S. Kvatinsky, "Memory processing unit for in-memory processing," in Proc. IEEE/ACM Int. Symp. Nanoscale Architectures (NANOARCH), July 2016, pp. 171–172.
- [65] R. B. Hur and S. Kvatinsky, "Memristive memory processing unit (MPU) controller for in-memory processing," in Proc. IEEE Int. Conf. Science Electrical Engineering (ICSEE), Nov. 2016, pp. 1–5.
- [66] P.-E. Gaillardon et al., "The programmable logic-in-memory (PLiM) computer," in Proc. IEEE Design, Automation & Test in Europe Conf. & Exhibition (DATE), 2016, pp. 427–432.
- [67] G. Papandroulidakis, I. Vourkas, A. Abusleme, G. C. Sirakoulis, and A. Rubio, "Crossbar-based memristive logic-in-memory architecture," *IEEE Trans. Nanotechnol.*, vol. 16, no. 3, pp. 491–501, May 2017.
- [68] R. B. Hur, N. Wald, N. Talati, and S. Kvatinsky, "SIMPLE MAGIC: Synthesis and in-memory mapping of logic execution for memristor-aided logic," in Proc. 36th Int. Conf. Computer-Aided Design. Piscataway, NJ: IEEE Press, 2017, pp. 225–232.

- [69] A. J. Lohn, J. E. Stevens, P. R. Mickel, and M. J. Marinella, "Optimizing TaO_x memristor performance and consistency within the reactive sputtering forbidden region," *Appl. Phys. Lett.*, vol. 103, no. 6, p. 063,502, 2013.
- [70] X. Lian, M. Wang, M. Rao, P. Yan, J. J. Yang, and F. Miao, "Characteristics and transport mechanisms of triple switching regimes of TaO_x memristor," *Appl. Phys. Lett.*, vol. 110, no. 17, p. 173,504, 2017.
- [71] H. Abunahla et al., "Switching characteristics of microscale unipolar Pd/Hf/HfO₂/Pd memristors," *Microelectron. Eng.*, vol. 185, pp. 35–42, 2018.
- [72] B. Jin-shun and H. Zheng-sheng, "Study on Hf/HfO₂ bipolar resistive random-access-memory," *J. Functional Mater. Devices*, vol. 5, p. 002, 2014.
- [73] H. Jiang et al., *Sci. Rep.*, vol. 6, p. 28,525, 2016.
- [74] S. H. Jo, K.-H. Kim, and W. Lu, "Programmable resistance switching in nanoscale two-terminal devices," *Nano Lett.*, vol. 9, no. 1, pp. 496–500, 2009.
- [75] A. Radoi, M. Dragoman, and D. Dragoman, "Memristor device based on carbon nanotubes decorated with gold nanoislands," *Appl. Phys. Lett.*, vol. 99, no. 9, pp. 093,102, 2011. doi: 10.1063/1.3633352.
- [76] A. Chanthbouala, V. Garcia, R. O. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, and M. Bibes, "A ferroelectric memristor," *Nature Materials*, vol. 11, pp. 860–864, 2012.
- [77] T. D. Dongale et al., "Development of Ag/ZnO/FTO thin film memristor using aqueous chemical route," *Mater. Sci. Semicond. Process.*, vol. 40, no. Suppl. C, pp. 523–526, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1369800115300408>
- [78] T. D. Dongale et al., "Development of Ag/WO₃/ITO thin film memristor using spray pyrolysis method," *Electron. Mater. Lett.*, vol. 11, no. 6, pp. 944–948, Nov. 2015. doi: 10.1007/s13391-015-4180-4
- [79] A. Ascoli, R. Tetzlaff, F. Corinto, and M. Gilli, "PSpice switch-based versatile memristor model," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, 2013, pp. 205–208.
- [80] D. Wheeler et al., "Fabrication and characterization of tungsten-oxide-based memristors for neuromorphic circuits," in *Proc. IEEE 14th Int. Workshop Cellular Nanoscale Networks and their Applications (CNNA)*, 2014, pp. 1–2.
- [81] K. Miller, K. S. Nalwa, A. Bergerud, N. M. Neihart, and S. Chaudhary, "Memristive behavior in thin anodic titania," *IEEE Electron Device Lett.*, vol. 31, no. 7, pp. 737–739, 2010.
- [82] O. Kavehei, K. Cho, S. Lee, S.-J. Kim, S. Al-Sarawi, D. Abbott, and K. Eshraghian, "Fabrication and modeling of Ag/TiO_x/ITO memristor," in *Proc. IEEE 54th Int. Midwest Symp. Circuits and Systems (MWSCAS)*, 2011, pp. 1–4.
- [83] L. Gao, P.-Y. Chen, and S. Yu, "NbO_x based oscillation neuron for neuromorphic computing," *Appl. Phys. Lett.*, vol. 111, no. 10, p. 103,503, 2017. [Online].
- [84] C. Funck et al., "Multidimensional simulation of threshold switching in NbO₂ based on an electric field triggered thermal runaway model," *Adv. Electron. Mater.*, vol. 2, no. 7, p. 1,600,169, [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aelm.201600169>
- [85] G. A. Gibson et al., "An accurate locally active memristor model for s-type negative differential resistance in NbO_x," *Appl. Phys. Lett.*, vol. 108, no. 2, p. 023,505, 2016. [Online].
- [86] J. P. Strachan et al., "State dynamics and modeling of tantalum oxide memristors," *IEEE Trans. on Electron Devices*, vol. 60, no. 7, pp. 2194–2202, July 2013.
- [87] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "VTEAM: A general model for voltage-controlled memristors," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 62, no. 8, pp. 786–790, 2015.
- [88] Z. Jiang et al., "A compact model for metal-oxide resistive random access memory with experiment verification," *IEEE Trans. Electron. Devices*, vol. 63, no. 5, pp. 1884–1892, May 2016.
- [89] K. Fleck, C. La Torre, N. Aslam, S. Hoffmann-Eifert, U. Böttger, and S. Menzel, "Uniting gradual and abrupt set processes in resistive switching oxides," *Phys. Rev. Appl.*, vol. 6, p. 064,015, Dec. 2016. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevApplied.6.064015>
- [90] A. Serb, J. Bill, A. Khiat, R. Berdan, R. Legenstein, and T. Prodromakis, "Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses," *Nature Commun.*, vol. 4, 2016.
- [91] Y. Pershin and M. D. Ventra, "Solving mazes memristors: A massively parallel approach," vol. 84, p. 046,703, Oct. 2011.
- [92] A. Ascoli, R. Tetzlaff, L. O. Chua, J. P. Strachan, and R. S. Williams, "History erase effect in a non-volatile memristor," *IEEE Trans. Circuits Syst. I, Regular Papers*, vol. 63, no. 3, pp. 389–400, Mar. 2016.
- [93] S. Ghedira, F. O. Rziga, K. Mbarek, and K. Besbes, "Dynamical resistive switching of a generic memristor model: Analysis and simulation," in *Proc. Int. Conf. Engineering MIS (ICEMIS)*, May 2017, pp. 1–5.
- [94] D. Radakovits and N. TaheriNejad, "Implementation and characterization of a memristive memory system," in *Proc. IEEE 32nd Canadian Conf. Electrical and Computer Engineering (CCECE)*, May 2019, pp. 1–5.
- [95] Knowm, "Knowm BS-AF-W memristor datasheet." 2017. [Online]. Available: <https://knowm.org/wp-content/uploads/DM8-16DIP-BS-AF-W.pdf>
- [96] Crossbar. (2017). *ReRAM*. [Online]. Available: <https://www.crossbar-inc.com/en/>
- [97] D. Bielek, M. Di Ventra, and Y. V. Pershin, "Reliable spice simulations of memristors, memcapacitors and meminductors," arXiv Preprint, arXiv:1307.2717, 2013.
- [98] B. K'A and E. E. Swartzlander, "Memristor-based arithmetic," in *Conf. Rec. 44th Asilomar Conf. Sig., Sys. and Comp.*, 2010.
- [99] S. Haghiri, A. Nemati, S. Feizi, A. Amirsoleiman, A. Ahmadi, and M. Ahmadi, "A memristor based binary multiplier," in *Proc. IEEE 30th Canadian Conf. Electrical and Computer Engineering (CCECE)*, Apr. 2017, pp. 1–4.
- [100] A. Karimi and A. Rezaei, "Novel design for a memristor-based full adder using a new imply logic approach," *J. Computational Electronics*, vol. 17, no. 3, pp. 1303–1314, Sept. 2018. <https://doi.org/10.1007/s10825-018-1198-5>
- [101] S. G. Rohani, N. TaheriNejad, and D. Radakovits, "A semi-parallel full-adder in imply logic," *Submitted to The IEEE Trans. on VLSI Systems*, June 2018.
- [102] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, "A memristor device model," *IEEE Electron Device Lett.*, vol. 32, no. 10, pp. 1436–1438, 2011.
- [103] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: Properties of basic electrical circuits," *Eur. J. Phys.*, vol. 30, no. 4, p. 661, 2009.
- [104] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: Threshold adaptive memristor model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 1, pp. 211–221, 2013.
- [105] Mar. 2018. [Online]. Available: <https://www.ict.tuwien.ac.at/staff/taherinejad/projects/memristor/files/VTEAM.sub>
- [106] S. H. Jo, K.-H. Kim, and W. Lu, "High-density crossbar arrays based on a Si memristive system," *Nano letters*, vol. 9, no. 2, pp. 870–874, 2009.
- [107] M. Hu, H. Li, Y. Chen, X. Wang, and R. E. Pino, "Geometry variations analysis of TiO₂ thin-film and spintronic memristors," in *Proc. IEEE 16th Asia and South Pacific Design Automation Conf. (ASP-DAC)*, 2011, pp. 25–30.
- [108] B. Jiao et al., "Resistive switching variability study on 1T1R AlOx/WOx-based RRAM array," in *Proc. IEEE Int. Conf. Electron Devices and Solid-State Circuits (EDSSC)*, 2013, pp. 1–2.
- [109] S. Yu, X. Guan, and H.-S. P. Wong, "On the stochastic nature of resistive switching in metal oxide RRAM: Physical modeling, Monte Carlo simulation, and experimental characterization," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, 2011, p. 17–3.
- [110] D. Niu, Y. Chen, C. Xu, and Y. Xie, "Impact of process variations on emerging memristor," in *Proc. 47th Design Automation Conf. ACM*, 2010, pp. 877–882.
- [111] P. Pouyan, E. Amat, and A. Rubio, "Statistical lifetime analysis of memristive crossbar matrix," in *Proc. IEEE 10th Int. Conf. Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, 2015, pp. 1–6.
- [112] A. C. Torrezan, J. P. Strachan, G. Medeiros-Ribeiro, and R. S. Williams, "Sub-nanosecond switching of a tantalum oxide memristor," *Nanotechnology*, vol. 22, no. 48, p. 485,203, 2011.
- [113] H. Lee et al., "Evidence and solution of over-RESET problem for HfO_x based resistive memory with sub-ns switching speed and high endurance," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, 2010, pp. 19–17.
- [114] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, and K. N. Salama, "Memristor-based memory: The sneak paths problem and solutions," *Microelectron. J.*, vol. 44, no. 2, pp. 176–183, 2013.
- [115] Y. Cassuto, S. Kvatinsky, and E. Yaakobi, "Sneak-path constraints in memristor crossbar arrays," in *Proc. IEEE Int. Symp. Information Theory Proc. (ISIT)*, 2013, pp. 156–160.
- [116] R. Berdan, A. Serb, A. Khiat, A. Regoutz, C. Papavassiliou, and T. Prodromakis, "A μ-controller-based system for interfacing selectorless RRAM crossbar arrays," *IEEE Trans. Electron. Devices*, vol. 62, no. 7, pp. 2190–2196, 2015.
- [117] J. K. Eshraghian, K.-R. Cho, H. H. Iu, T. Fernando, N. Iannella, S.-M. Kang, and K. Eshraghian, "Maximization of crossbar array memory using fundamental memristor theory," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 64, no. 12, pp. 1402–1406, 2017.



Applications of Deep Learning to Audio Generation

Yuanjun Zhao, *Student Member, IEEE*, Xianjun Xia, *Student Member, IEEE*,
and Roberto Togneri *Senior Member, IEEE*

Abstract

In the recent past years, deep learning based machine learning systems have demonstrated remarkable success for a wide range of learning tasks in multiple domains such as computer vision, speech recognition and other pattern recognition based applications. The purpose of this article is to contribute a timely review and introduction of state-of-the-art deep learning techniques and their effectiveness in speech/acoustic signal processing. Thorough investigations of various deep learning architectures are provided under the categories of discriminative and generative algorithms, including the up-to-date Generative Adversarial Networks (GANs) as an integrated model. A comprehensive overview of applications in audio generation is highlighted. Based on understandings from

these approaches, we discuss how deep learning methods can benefit the field of speech/acoustic signal synthesis and the potential issues that need to be addressed for prospective real-world scenarios. We hope this survey provides a valuable reference for practitioners seeking to innovate in the usage of deep learning approaches for speech/acoustic signal generation.

I. Introduction

Since the phonograph was invented in 1877 by Edison, different types of acoustic sounds like voice, music and birdsong can be preserved as audio recordings [1]. After decades of exploration and analysis, essential properties of various acoustic signals have been revealed for a deep understanding of the mechanism for production, propagation and perception. The diverse

Digital Object Identifier 10.1109/MCAS.2019.2945210

Date of current version: 19 November 2019

and complex information contained in acoustic signals is useful for spatial location, determining the species of a sound source, identity of a speaker and the message being conveyed [2]. Ever since the advantages of digital signal processing and machine learning technology were consolidated into acoustic signal processing systems, the effectiveness of algorithms that can describe, categorize and interpret all manner of sounds has been improved significantly. Recently, the focus of research in machine learning algorithms has moved onto more practical and emerging fields like speech/speaker recognition, speech synthesis and acoustic event/scene recognition. In this paper, a thorough survey of state-of-the-art deep learning algorithms applied to Speech Synthesis (SS) and Voice Conversion (VC) will be provided. Other types of acoustic signals like music and singing will also be included. However the analysis can be used for reference to any form of audio generating system or application.

The task of speech synthesis is to build natural-sounding synthetic voices with either knowledge-based or data-based approaches [3]. A typical text-to-speech (TTS) system is shown in Fig. 1. In the front-end, the raw texts are firstly converted into the equivalent of written-out words for the task of text normalization. Then the phonetic transcriptions are assigned to each word and the texts are marked into prosodic units. This step is usually called text-to-phoneme conversion. The output of the front-end is the symbolic linguistic representation. The back-end of the TTS system is referred to as the synthesizer. The task of the synthesizer is to convert the symbolic linguistic representation into sound.

Conventional speech synthesis technology are diphone synthesis, concatenative speech synthesis and statistical parametric speech synthesis (SPSS). Unit selection based concatenative techniques have been the main approaches to speech synthesis, of which the quality of the generated speech depends on the avail-

able corpora. Suitable sub-word units are automatically chosen from selected corpora of natural speech [4]. Unit selection based toolkits, like the open-source multilingual TTS synthesis platform MaryTTS [5], have been embedded in commercial applications and can bring synthetic speech with a high level of quality. In contrast to retain unmodified speech components in unit selection based methods, statistical parametric speech synthesis systems use parametric models to generate universal descriptions of speech subsets with similar sounding segments. Specifically, the speech are described by models using parameters instead of stored exemplars. These parameters are represented by statistics such as means and variances of probability density functions found in the training data. One of the most popular statistical parametric synthesis techniques is the Hidden Markov Model (HMM) synthesis, which allows more variations on the speech data by statistically modeling and generating the speech parameters of a speech unit with the maximum likelihood criterion based HMMs [6]. An annual challenge named the Blizzard Challenge¹ is held to compare research techniques in corpus-based speech synthesizers, which advances the better understanding and exploration of effective speech synthesis methods.

Compared to the speech synthesis, the voice conversion is a different technique to create high-quality synthetic speech. The task of voice conversion is to convert a sentence spoken by an original speaker to a resultant utterance with the same sentence as before. The resultant utterance sounds as being spoken from a different speaker. The information of the timbre and the prosody of the original and target speakers are considered in a voice conversion system. These two features are generally associated with the dynamic spectral envelope of the voice signal, pitch/energy contours and rhythmic distribution of phonemes [7]. A diagram of the typical voice conversion system is given in Fig. 2. In the training phase, the speech signal from the source speakers and the target speakers are fed into the VC system. After the speech analysis and the mapping feature computation, the raw speech signals are converted into a suitable representation for the further processing and modification. The speech segments of the training signals are aligned with respect to time. The conversion function is then trained on these aligned mapping features. In the conversion phase, the mapping features of a new input utterance are also extracted and then converted by the

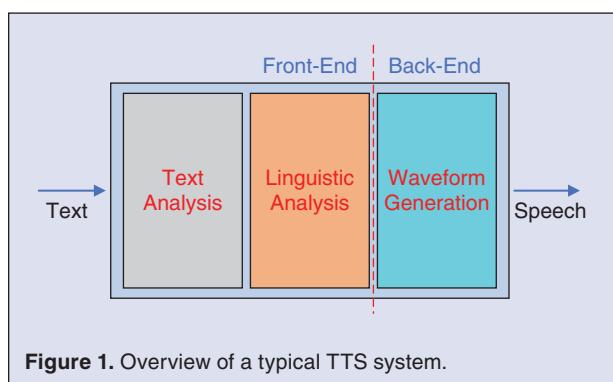


Figure 1. Overview of a typical TTS system.

¹<http://www.festvox.org/blizzard/blizzard2017.html>.

trained conversion function. After the conversion step, the speech features are computed from the converted features and used for synthesizing the converted utterance waveform.

Since the original formulation of the voice conversion problem was proposed in 1985, many VC techniques have been introduced [8]. The most popular VC approach is the STRAIGHT (Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum), which utilizes a combination of spectral representations and voice conversion techniques [9]. With this VC system, the spectral, acousti-

cal and rhythmic parameters can be manipulated easily. Via suitable modification of the speech waveform, voice conversion can convert non-/para-linguistic information as expected [10]. Table I lists several existing frameworks that are frequently applied for producing speech and acoustic signals. Details of speech synthesis and voice conversion techniques can be found in the overview papers and books of [3], [6], [7], [11].

In recent years, the research on anti-spoofing countermeasures to detect deliberate and unknown attacks by artificial (e.g. synthesized) speech is becoming more active [12]. Compared to other state-of-the-art

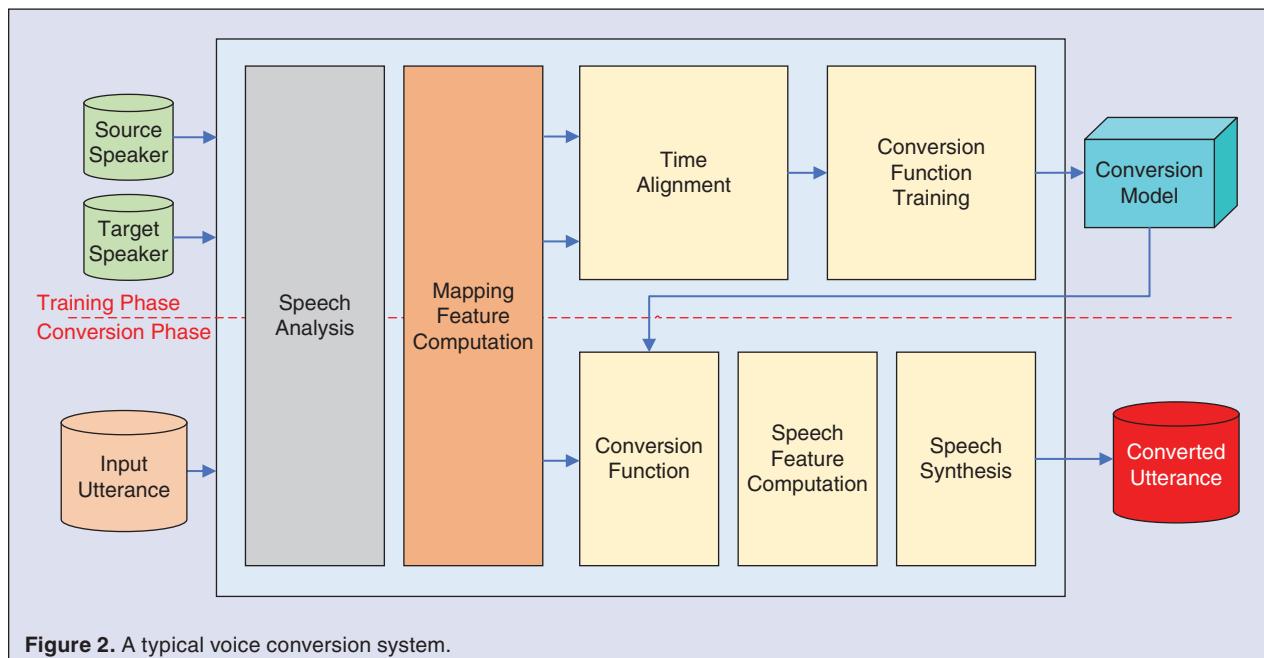


Table I.
Frameworks providing speech generation tools and methods.

| Framework | Language | URL |
|------------------|-------------------|---|
| MaryTTS | Java | http://mary.dfki.de/ |
| eSpeak | C | http://espeak.sourceforge.net/ |
| HTS | C | http://hts.sp.nitech.ac.jp/ |
| Festvox/Festival | C++ | http://festvox.org/ |
| Idlak | C++ | https://github.com/idlak/idlak |
| Merlin | Python | https://github.com/CSTR-Edinburgh/merlin |
| Ossian | Python | https://github.com/CSTR-Edinburgh/Ossian |
| GAN TTS | Python | https://github.com/r9y9/gantts |
| Sprocket | Python | https://github.com/k2kobayashi/sprocket |
| STRAIGHT | Matlab | https://github.com/HidekiKawahara/legacy_STRAIGHT |
| WORLD | C++/Matlab/Python | https://github.com/mmorise/World |

anti-spoofing methods like [13], [14], deep learning based approaches provide improved capability for protecting automatic speaker verification (ASV) systems from speaker spoofing via synthetic speech [15].

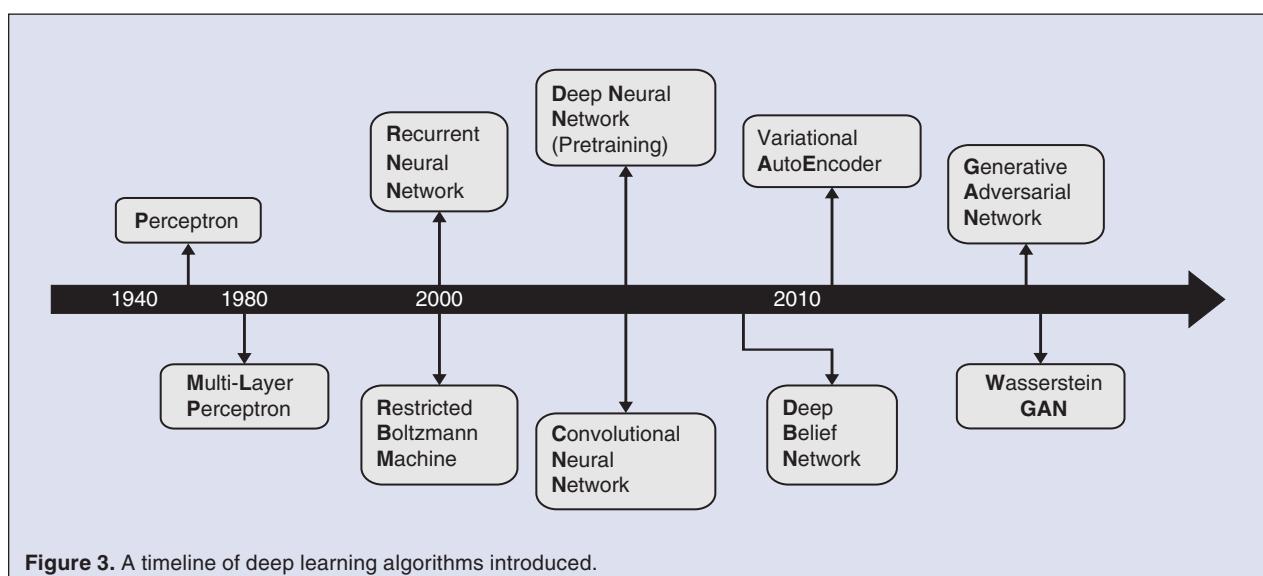
Besides the acoustic signals produced by human vocal systems (e.g. speech and song), another ultimate acoustic ‘language’ is that of music. To compose a natural-sounding music, several types of musical content need to be generated, such as the melody, polyphony, counterpoint, chords and lead sheet. By a combination of all these musical components, music, which is well-known for its emotional effect, can act as a form of expression and precisionness [16]. More details about music generation can be found in refs. [17], [18].

Machine learning has been the dominating technology in a variety of signal processing applications. In processing raw data from the natural information source, conventional machine learning approaches are limited by the complicated requirements of feature extraction [19]. Deep learning (DL), as a subfield of machine learning, involves a hierarchical computing framework in which multiple layers of learning algorithms are stacked in a specific order. With an approximation of non-linear functions, deep learning algorithms are applied to achieve abstract representations and learn feature vectors from the original data. We have seen impressive progress of deep learning over the last decade, to many engineering and science application areas ranging from web searches, online content filtering and recommendations systems to computer vision, speech recognition and other machine intelligence tasks [20].

In Fig. 3 we depict the historical development of the deep learning algorithms discussed in this paper. The artificial neural network (ANN) [21] originated in the 1940s

and led to the first wave of artificial intelligence (AI) algorithms with the creation of the single-layer perceptron (SLP) and the multi-layer perceptron (MLP) [22], [23]. To build a standard neural network (NN), varying amounts of neurons are used to yield real-valued activations and, by adjusting the weights and biases, the NNs can behave as expected in specific tasks like computer vision and speech recognition [24]. The development of ANNs stagnated because of the incapability of processing the exclusive-or circuit and computing devices with low processing capacity. The next wave of research boomed in the 1980s when the backpropagation algorithm (BP) [25] was proposed. As an efficient gradient descent algorithm, BP effectively accelerated the training of ANNs. However, in the late 1990s, ANNs and the BP algorithm were largely abandoned by the community. It was generally believed that backpropagation would get trapped in poor local minima and the average error could not be reduced any more. In addition to that, insufficient labeled training data can cause the problem of over-fitting [19].

In 2006, which is considered the first year of deep learning, several breakthroughs regarding new network architectures and training methods revived the interest in neural networks. In [26], a new layer-wise-greedy-learning based method was proposed for training very deep neural networks (DNN). Hidden layers in a network are pre-trained one layer at a time using the unsupervised learning approach and this considerably helps to accelerate subsequent supervised learning through the backpropagation algorithm [19], [27]. Also in 2006, a Convolutional Neural Network (CNN) trained by BP set a new record of 0.39% on the handwriting digits database MNIST [28], which was a significant progress in the performance since the classical prototype LeNet-5 [29].



After the ImageNet challenge² in 2012, CNNs began to rule the field of image classification [30]. New innovations have been made every year and the research in CNNs has proliferated. The AlexNet, as the winner of 2012 challenge, was considered to be the break-through CNN variant with a top-5 test error rate (a score used to indicate if the target label is one of the top 5 predictions with the highest probabilities) of only 17.0% on ILSVRC-2010 test set [31]. In 2013, the ZF Net gave a better accuracy based on a slight modification of the AlexNet model and proposed a novel feature maps visualization [32]. Another AlexNet type architecture based CNN is the VGG Net and it was frequently applied in the community because of its appealing uniform network architecture [33]. Among all the submissions, the GoogLeNet proposed by Google won the 2014 ImageNet challenge [34]. In GoogLeNet, the core innovation is a repeated inception module, with which the number of the parameters can be significantly reduced. The next popular CNN is the Microsoft ResNet (residual network) which was the winner of the 2015 ImageNet challenge [35]. The residual block was proposed in this network to guarantee the gradients computed can be directly used into a more effective weights updating process.

The Autoencoders (AEs) are another popular type of unsupervised pre-training deep feedforward neural network (FNN) [36]. Different variations of AEs have been introduced to enhance the ability of extracting informative representations, including the denoising autoencoder [37], [38], the sparse autoencoder [39], the variational autoencoder (VAE) [40] and the contractive autoencoder (CAE) [41].

In speech/acoustic recognition and other pattern recognition problems, generative and discriminative models are the two main approaches. The generative models attempt to provide the distribution of data or the joint distribution of data and the corresponding targets, while the discriminative models directly predict the distribution of targets conditioned on the data [42]. The Generative Adversarial Networks (GANs) proposed by Goodfellow can be considered as a hybrid model consisting of both the generative and the discriminative models [43]. In the GAN's framework, a generator G and a discriminator D are trained simultaneously to capture the data distribution and to estimate the probability that an output is from the training data.

With the rapid development of efficient computation techniques and the growth of computing power, implementing large-scale deep learning approaches with computers is no longer a fantasy. The advent of fast Graphic Processing Units (GPUs) and the avail-

ability of huge amounts of data significantly expedite the training and fine-tuning of deep learning models. For speech synthesis, deep learning aims to remove the restrictions of the conventional method in statistical parametric synthesis based on Gaussian-Hidden Markov Models and other classical models [44]. A properly designed acoustic model plays an important role in HMM-based statistical parametric speech synthesis systems. Generated speech from shallow-structured HMM-based synthesizers have been generally known for poor fidelity compared with natural speech. To address this problem, deep learning approaches are adopted to offset this deficiency. Generative models like Restricted Boltzmann Machines (RBMs) and Deep Belief Networks (DBNs) have displaced traditional Gaussian models and concrete improvements have been obtained with regard to the quality of speech [45], [46]. In contrast to the generative deep models, discriminative models of the DNN can also be applied for performing speech synthesis. In [47], a DNN-based approach was proposed to predict spectral and excitation parameters. A better quality of synthetic speech was achieved with a similar number of parameters to the HMM-based synthesizer. For voice conversion, deep learning is usually used for learning the associations between the spectral mapping features, which allows modification of speech properties in a VC system. In [48], a two-layer feedback neural network based on bidirectional associative memory was reformulated to model the spectral envelope space of the speech signal. Experimental results showed that the proposed method has a better modeling ability than the traditional use of Gaussians with diagonal covariance. In [49], a stacked joint-autoencoder was applied to construct a regression function, which was used in a VC task. This method demonstrated features that do not suffer from the averaging effect inherent with the back-propagation algorithm.

As a fast growing domain, deep learning has recently been applied for musical audio synthesis and music composition. A fundamental motivation is to learn a representation of diverse musical styles or musical content based on several widely available databases. Generally, research works focus on four main dimensions of music generation: the objective (e.g., a melody or a sequence of chords), the representation (e.g., raw musical signal or spectral features), the architecture (e.g., DNNs, CNNs or RNNs) and the learning strategy [17]. In [50], an autoencoder based synthesizer was proposed for compressing and reconstructing magnitude short time Fourier transform frames. This synthesizer was re-trained for music synthesis applications and several

²<http://www.image-net.org/>.

samples were generated.³ In [51], an interactive musical audio synthesis system⁴ based on ANNs was introduced. In this system, frames of low-level features were learned by an ANN and a high level representation of the musical audio was learned through an autoencoder.

The purpose of this paper is to offer a timely overview of the applications of deep learning approaches to audio generation problems, especially speech generation. Deep learning algorithms involved are categorized as either discriminative or generative methods, taking GANs as a distinctive hybrid method. The background on the different deep learning architectures and up-to-date applications are provided to readers in this area. The rest of the paper is organized as follows. In Section II we start by reviewing recent deep discriminative algorithms and the remarkable progress

in speech synthesis and acoustic signals generation. While the same understanding and analysis are shown for deep generative algorithms in Section III. Specific discussion of GANs and their variations are reported in Section IV. The reasons why deep learning can be beneficial for pattern recognition problems and issues to be studied further are given in Section V. We conclude the paper in Section VI.

II. Discriminative Algorithms

The distinction between discriminative and generative models is the probability distribution modeled. Generally, an output variable \mathbf{y} needs to be estimated by a standard pattern recognition model given an input variable \mathbf{x} . A deep discriminative algorithm, like a DNN, utilizes multi-layer hierarchical architectures to directly compute the probability of \mathbf{y} given an \mathbf{x} , i.e. to estimate $p(\mathbf{y}|\mathbf{x})$. The most common discriminative models used in machine learning include logistic/linear regression, support vector machines (SVMs), random forests and neural networks. With different structural elements, deep discriminative models contain various implementations in terms of tasks and functions. In this paper we only focus on the discriminative models with deep hierarchical architectures such as the MLP, CNNs and RNNs.

A. Deep Discriminative Architectures

1) Multi-Layer Perceptron

A multi-layer perceptron, as a type of feedforward artificial neural network, consists of at least three layers of computing units (also known as “nodes” or “neurons”). A diagram of a single layer MLP is shown in Fig. 4. Except for an input and an output layer, one or more hidden layers can be inserted as required. Each node in a hidden layer or output layer contains a nonlinear activation function depending on how the network has been configured. The computation process of the l th layer is shown in Fig. 5 and is described by:

$$y^l = f^l(\mathbf{w}^l \mathbf{x}^l + b^l) \quad (1)$$

where \mathbf{x} are inputs to the network while weights and biases are denoted as \mathbf{w} and b , respectively. The activation function f is nonlinear to enhance the capacity of the modeling representation, especially beneficial for nonlinear classification problems. Like in [26], the back-propagation algorithm is employed to update weights and biases of all the layers with the loss calculated. Specific objective functions are set to estimate the distance between the prediction and the actual value. With trained network parameters, we can use the MLP to

³http://soundcloud.com/ann_synth

⁴<https://github.com/woodshop/deepAutoController>

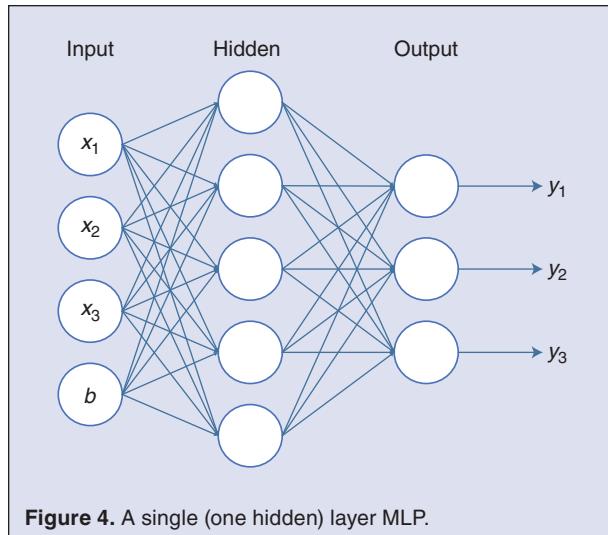


Figure 4. A single (one hidden) layer MLP.

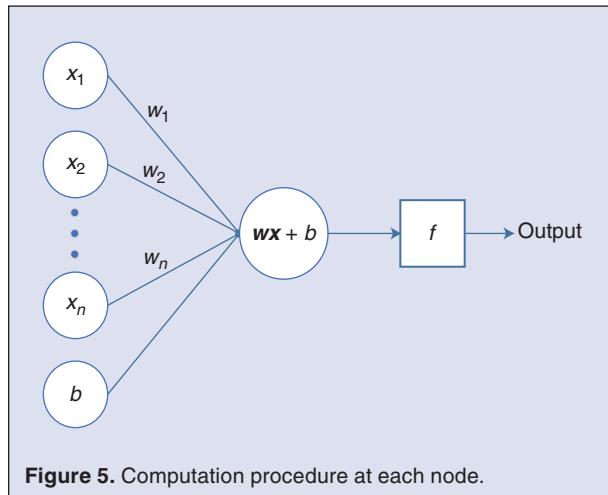


Figure 5. Computation procedure at each node.

predict the output in practical pattern recognition problems. More details of the backpropagation algorithm can be found in [52].

2) Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have been usually exploited in the fields of image classification, object detection, face recognition and speech processing. Compared with general DNNs, matrix multiplication is replaced by the convolution to dramatically reduce the amount of weights in modeling. This makes CNNs a powerful tool to relieve the memory curse in image processing related tasks since the large number of pixels in a single picture always leads to a similarly large number network parameters. Consequently, the complexity of the network can be decreased. Another advantage of CNNs is that the images, as raw inputs, can be directly imported to the network without the feature extraction procedure. In addition, with a workstation equipped with GPUs, training a CNN is more efficient.

Generally, a basic CNN includes three types of layers: convolutional, pooling and fully-connected layers. For example, Fig. 6 demonstrates the famous LeNet-5. In this network, there are three convolutional layers, two pooling layers and two fully-connected layers. For each convolutional layer, there are several convolution kernels to compute feature maps from the previous layer. Distinct feature representations of original inputs can be learned by the convolution procedure and transferred to subsequent layers. A region of neighboring neurons are mapped to single neuron on the subsequent layer and this region is referred to as the receptive field of this single neuron. To achieve a feature map, the input is convolved by a trainable kernel and the result is ap-

plied with an element-wise nonlinear activation function. Due to the property of parameter sharing in CNNs, the kernels are shared by all pixel areas of the input. Finally, all the feature maps are obtained by utilizing a variety of convolutional kernels. The pooling layers are introduced to reduce the overall size of the signal to avoid over-fitting problems caused by high dimensional inputs. As in classical neural networks like the MLP, neurons in all types of layers compute the dot product between the input vectors and the weights, to which biases are added. Then the weighted sum is passed through a nonlinear activation function to obtain the output vectors. Denoting the pooling function as $\text{pool}(\cdot)$, for each feature map we have:

$$y_{i,j,k}^l = \text{pool}(f_{m,n,k}^l(\mathbf{w}_k^T \mathbf{x}_{m,n}^l + b_k^l)), \forall (m,n) \in \mathfrak{R}_{ij} \quad (2)$$

where \mathfrak{R}_{ij} is a local neighborhood around location (i,j) in the k th feature map of the l th layer [53]. Note that the weights and the bias are shared by the neurons belonging to the same layer. From (1) and (2) it is obvious that the changes between the classical NN architectures and the CNNs are only in the layer category and configuration.

High-level feature representations can be extracted gradually by stacking several convolutional and pooling layers. The one or more fully-connected layers added before the output layer aim to learn nonlinear combinations of the high-level features passed from the previous layers and convey the higher-level representation to output layers. Fully-connected layers can be optionally replaced by convolution layers, of which the size of the convolution kernels is 1×1 [54]. The last layer in a CNN is known as the output layer, which is configured based on the type of task that the network needs to

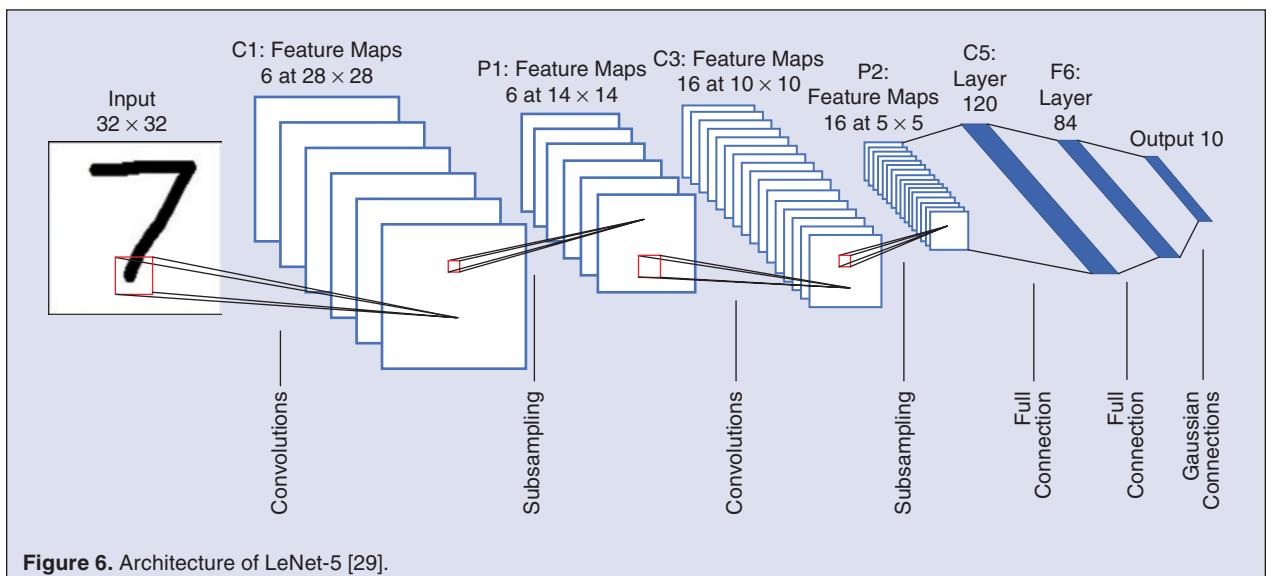


Figure 6. Architecture of LeNet-5 [29].

perform. The softmax operator is a commonly used output layer for classification tasks, of which the resultant vectors present a classification probability distribution summing to 1. For training a CNN, the goal is to minimize the objective function defined for the task. With an appropriate optimizer, the best fitting set of parameters (the weights and bias terms) can be solved.

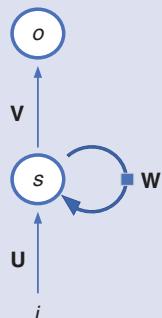


Figure 7. A simple computation unit of RNNs.

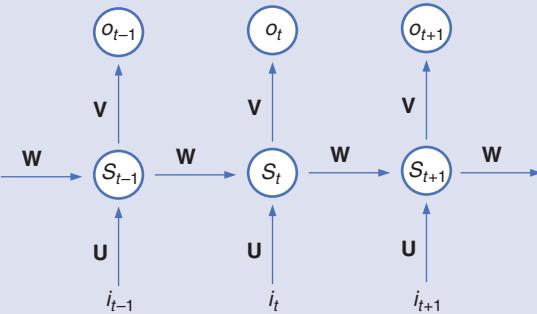


Figure 8. Unfolded structures of RNNs.

3) Recurrent Neural Networks

Recurrent neural networks (RNNs) are developed to process pattern recognition tasks of sequential data like text, genomes, spoken words or numerical times series data from real-world commerce [55]. Compared to the MLP and CNNs, RNNs explore the temporal information of inputs by taking time and sequence into account. To be specific, in the case of feedforward networks as shown in Fig. 4, samples fed to the network are transformed straight into an output via supervised learning. While in RNNs, the output at time step $t - 1$ affects the future decision one moment later at time step t by the feedback loops. This unique component makes RNNs have “memory,” which is indispensable for language processing and sequence signal processing.

Fig. 7 indicates the basic schematic diagram in RNNs which is an integration of three kinds of layers: the input layer, the hidden layer and the output layer. The i denotes the input vector and o denotes the output vector. The hidden neurons are denoted as the vector s while \mathbf{U} , \mathbf{V} and \mathbf{W} are the weights matrix. Note that \mathbf{W} contains the information of the previous time steps for this neuron. After unfolding, the diagram of this basic RNN unit is represented as in Fig. 8. From Fig. 8, it is explicitly shown that the value of hidden layers depend on both the inputs and the hidden neurons of the previous time steps. The output and hidden layers can be calculated as below:

$$\begin{aligned} o_t &= g(Vs_t) \\ s_t &= f(Ui_t + Ws_{t-1}) \end{aligned} \quad (3)$$

For language processing, it is necessary to model not only previous words or phrases in a sentence, but also subsequent linguistic units at time step $t + 1$. Bi-directional RNNs are proposed to predict or label each element of a sequence with its past and future contexts.

Outputs of two RNNs can be concatenated to process the sequence from opposite directions. Fig. 9 is a graph demonstrating an example of such bi-directional RNNs.

In practical applications, it is found that the performance of the RNNs degrades when data sequences are long. Exploding gradients and vanishing gradient problems result in an uncontrolled propagation of the gradients in the training procedure. Presetting a threshold is an effective way to avoid the exploding gradients issue, but the vanishing gradient is tricky to restrain. Three methods

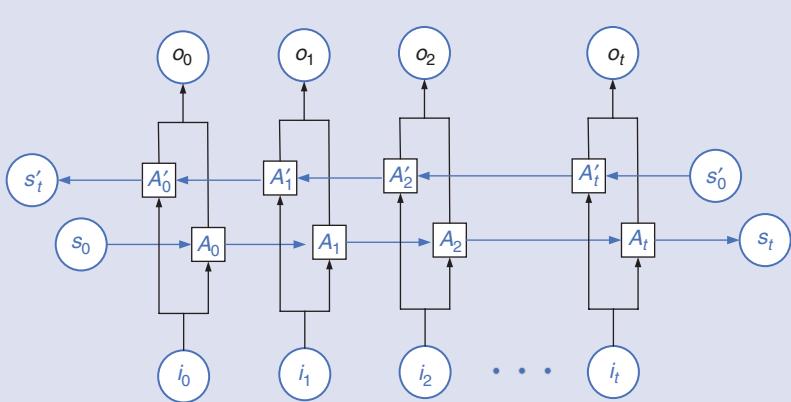


Figure 9. A simple bi-directional RNN.

have been adopted to optimize the RNN training process. One approach is to initialize the weights of RNNs in a proper manner to eliminate vanishing gradient issue. Or we can replace common activation functions (e.g. sigmoid or tanh) with rectified linear units (ReLU) [56], [57]. Deploying improved RNN structures, such as the Long Short Term Memory (LSTM) network [55], [58], [59] or the Gated Recurrent Unit (GRU) [60]–[62], has become the preferred solution.

In the following, selected applications of deep discriminative algorithms in speech and audio signal generation are reviewed.

B. Applications

1) Speech Generation

Deep discriminative models based methods are actively being investigated for acoustic modeling and adaption, feature learning and waveform generation in speech generation. For acoustic modeling, DNNs are used to replace the Gaussian mixture models (GMMs) for the evaluation between frames of acoustic observations and HMM states [63]. In [64], a mixture density network (MDN) was used as an acoustic model in statistical parametric speech synthesis. MDNs give full probability density functions over real-valued output features conditioned on the corresponding input features. This approach addressed the restrictions of the lack of ability to predict variances and the unimodal nature of the objective functions in statistical parametric speech synthesis. DNNs can also be used to replace the decision trees in HMM-based statistical parametric speech synthesis applications [47]. This alternative scheme assisted to break the limitations in the conventional decision tree-clustered context-dependent HMM-based approach, such as the inefficient expression of complex context dependencies and fragmented training data. In [65], DNN-based expressive speech synthesis was comprehensively investigated, especially for multiple emotions representations. In [66], a proof-of-concept system for speech texture synthesis and voice conversion was introduced. A cost function with respect to the input waveform samples was optimized in this system. Realistic speech babble and utterance in a different voice can be reconstructed.

LSTM based networks are also demonstrated for their surprising performance in statistical parametric speech synthesis systems. In [67], a simplified LSTM architecture was proposed and can achieve similar performance in speech synthesis with fewer parameters than vanilla LSTM. Another LSTM RNN was introduced in [68], which utilized data from multiple languages and speakers. Experimental results showed that this multi-

lingual statistical parametric speech synthesis system can generate speech in multiple languages from a signal model and the naturalness is satisfactory. SampleRNN, as a state-of-the-art RNN based model, was proposed in [69] for unconditional audio generation. This model is able to capture underlying sources of variations in the temporal sequences over very long time spans. The samples generated by the SampleRNN are preferred by human raters. In [70], RNNs with bi-directional LSTM (BLSTM) cells were adopted to capture the correlations between any two instants in a speech. This hybrid system of DNN and BLSTM-RNN can outperform both the traditional HMM-based and the more recent DNN-based statistical parametric speech synthesis systems with a smoother speech trajectory. The use of the deep BLSTM-RNN was also investigated in [71] for voice conversion. A sequence-based conversion method was proposed to improve the naturalness and the continuity of the converted speech. Deep BLSTM-RNNs were used to model both the frame-wise relationship between source/target voice and the long-range context-dependencies in the acoustic trajectory. The resultant speech showed a better MOS performance than DNN based methods.

2) Other Types Of Audio Signal

Recently, deep neural networks have been applied in music generation to meet the demands of music composition on various application platforms. In [72], an end-to-end melody and arrangement generation framework was proposed. This framework can generate melody tracks with several accompanying tracks played by diverse types of instruments via applying a multi-instrument co-arrangement model. In [73], a set of parallel, tied-weight recurrent networks was trained to model the polyphonic music. Two modified architectures were proposed and combined for the music generation and prediction task. Experimental results⁵ demonstrated that the proposed models can reproduce complex rhythms, melodies and counterpoints in some cases. LSTMs were adopted in [74] for the generation of polyphonic music. In this work, a chord LSTM was used to predict a chord progression based on a chord embedding. Then another LSTM was used to generate polyphonic music from the predicted chord progression. The produced music had a clear long-term structure that sounds as harmonic as the ones played by a musician.

In [75], a gated PixelCNN [76] was applied in a singing synthesizer. The harmonic spectral envelope was modeled by the network. In [77], the relationship between

⁵<https://www.cs.hmc.edu/~ddjohnson/tied-parallel/>

the musical score and its acoustic features was modeled in frames by a DNN. Subjective experimental results demonstrated that the DNN-based singing voice synthesizer outperformed the conventional HMM-based system in terms of naturalness. In listening tests, the deep learning based singing synthesizer can generate a sound quality on-par or exceeding the previous state-of-the-art concatenative methods. The timbre and fundamental frequency carried in the natural songs can be jointly modeled by deep learning approaches, which allows for much faster experimentation because of the superior tools like GPUs [78]. Another interesting application is the conditional rhythm composition [79], in this work a deep network was proposed as a combination of LSTMs and feed forward layers. This architecture was applied to learn long drum sequences from the metrical information and bass lines.

III. Generative Algorithms

For generative algorithms, they provide the joint probability distribution of the input and the output of the deep learning model. In other words, generative models aim to estimate $p(\mathbf{x}, \mathbf{y})$. However, $p(\mathbf{y}|\mathbf{x})$ can also be obtained with Bayes' theorem to indirectly perform pattern classification tasks [42]. Several widely used generative algorithms are Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), Restricted Boltzmann Machine (RBM), Deep belief Network (DBN) and Variational autoencoders (VAEs).

A. Deep Generative Architectures

I) Variational Autoencoder

The variational autoencoder (VAE) [40] is derived from the original autoencoder but with stronger assumptions concerning the distribution of latent variables. In

order to understand the VAEs we need to first introduce the autoencoder.

An autoencoder is a type of NN that learns data codings in an unsupervised manner [80]. Dimensionality reduction is the main application of autoencoders due to the capability of representing a set of data [36]. Fig. 10 is a classical schematic structure of an autoencoder with three fully-connected hidden layers.⁶ This is identical to the MLP which is built by many single layer perceptrons as a feedforward neural network. Note that for autoencoders, the number of nodes on the output layer is the same as the input layer. Generally, the size of the hidden layers are smaller than that of the input or output layers. This unique type of architecture is designed for compressing inputs to a compact code and uncompressing that code into the outputs. The outputs are expected to be closely matched to the inputs. The two main components of an autoencoder for completing the compress and the uncompress operations are called the encoder and the decoder, respectively. The output layer of the encoder (that is also the input layer of the decoder) is usually referred to as a code, latent variables or a latent representation. In pattern recognition tasks like image processing, the latent representation can be used as extracted features for a further supervised learning. While in unsupervised learning tasks, variations of the autoencoder (such as VAEs) can be used to generate new data from the training inputs. Therefore, autoencoders have emerged as one of the most popular approaches in the field of machine learning in recent years [81]–[85].

Although standard autoencoders have been successfully applied, there are several problems limiting the development of autoencoders. The most fundamental one

⁶The cat image used in Fig. 10 is from the YouTube video: <https://www.youtube.com/watch?v=YCaGYUlfidy4>

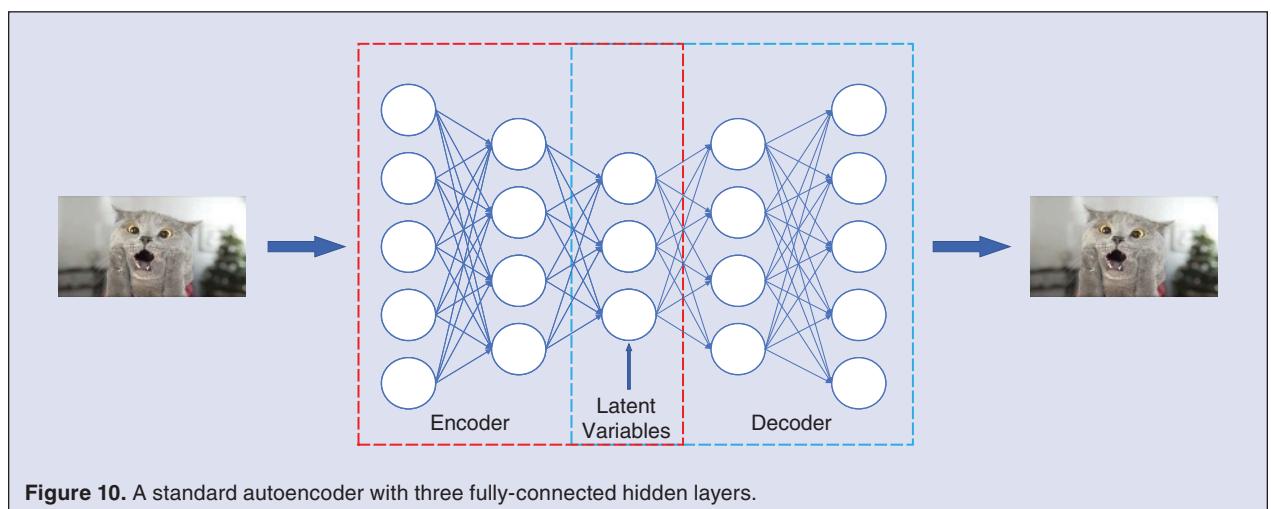


Figure 10. A standard autoencoder with three fully-connected hidden layers.

is that the latent space for generation may not be continuous and easily interpolated. This makes it difficult to build a generative model and produce variations in the input image, caused by a discontinuous region from the latent space.

Compared to the standard autoencoders, VAEs are able to generate new outputs which are similar to the training data in a desired direction. The latent spaces of VAEs are continuous to allow random sampling and interpolation by design. To achieve this unique property, the output of the encoder (i.e. the latent variables) is transformed to two vectors as shown in Fig. 11: one for the means μ and another for the standard deviations σ . These statistical parameters form a vector of random variables \mathbf{X} , of which the probability distribution of all elements X_i is normally distributed. The length of this random variable vector is the same as the encoding vector in standard autoencoders. The i th element of μ and σ are the mean and standard deviation of the i th random variable X_i . After sampling, a real-value vector is obtained and passed forward to the decoder for generation. Intuitively, the latent representation of an input is controlled by the mean and standard deviation vector. Consequently, new samples can be created by the decoder with a slightly varied latent encoding.

To achieve ideal latent variables, which are as close as possible to each other while still being distinct, the Kullback-Leibler (KL) divergence is introduced into the loss function. KL divergence is used to measure by how much two probability distributions diverge from each other. A lower KL divergence means the probability distribution of the generated samples are closer to the target distribution. For VAEs, the KL loss in a Gaussian case, as given in (4),

$$D_{KL}(Q \parallel P) = \frac{1}{2} \sum_{i=1}^n (\sigma_i^2 + \mu_i^2 - \log(\sigma_i^2) - 1) \quad (4)$$

is the sum of all the KL divergences between the approximate posterior Q and the standard normal distribution P [40]. Obviously, the minima will be reached when $\mu_i = 0$ and $\sigma_i = 1$. Besides the KL loss, the reconstruction loss (such as the mean square error) is also adopted to maintain the similarity of nearby latent variables.

A VAE can also be extended for supervised learning, namely as the Conditional VAE (CVAE). The CVAE models latent variables and data, both conditioned on some random variables, which allows producing data with specific attributes. More details can be found in [86].

2) Deep Belief Network

The deep belief network (DBN) is a type of generative graphical model and is usually considered as a stack

of simple restricted Boltzmann machines (RBMs). The edges between layers of a DBN are both directed and undirected. Hidden units are connected to those of other layers but not with units belonging to the same layers.

To understand DBNs, the definition of RBMs is introduced first. As their name implies, RBMs are derived from Boltzmann machines, which are stochastic recurrent neural networks. The difference is that in an RBM all the neurons are paired from two groups of units: visible and hidden units. In addition, there are no connections between neurons within a group. This is for a more efficient training algorithm, in particular the gradient-based algorithm. Fig. 12 shows an example of an RBM. It is noted that the connections between the visible layer and the hidden layer are bidirectional. The contrastive divergence (CD) algorithm is most often used to train an RBM by optimizing the weight matrix [87]. After training, the hidden layer can perform an exact feature representation of the visible layer and reconstruct the visible layer.

Returning back to the DBN, which is a combination of multiple RBMs, this composition leads to a fast, layer-by-layer unsupervised training procedure. The network architecture of a DBN is shown in Fig. 13. When the first RBM is trained by the CD algorithm, another RBM is stacked, taking the outputs from the hidden layer of the

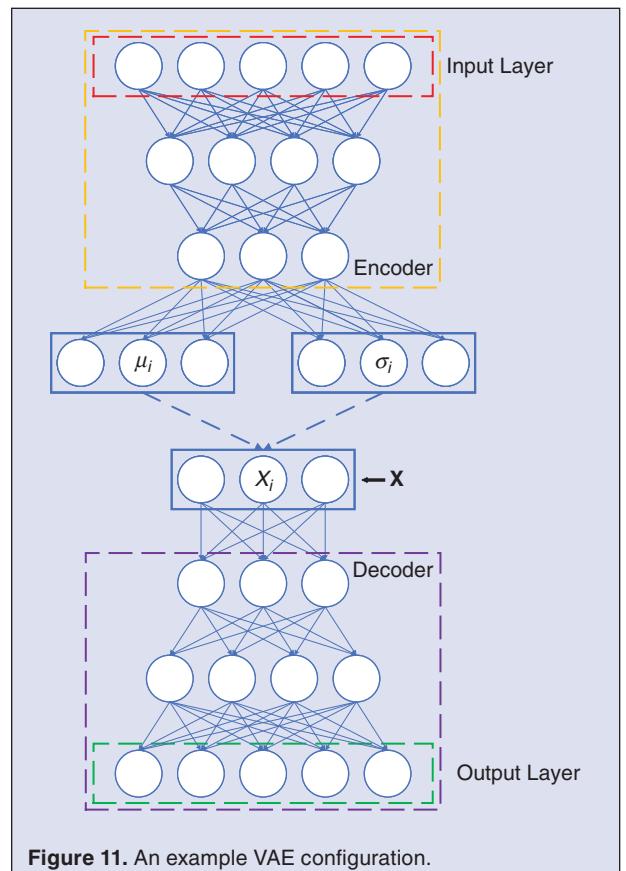


Figure 11. An example VAE configuration.

first RBM as the input. New weights and biases are assigned for the new visible layer to train the new RBM by the CD algorithm again. This process is iterated until reaching a desired stopping criterion.

Note that DBNs can be applied for the purpose of supervised learning by adding a final layer of variables that represent the desired outputs and backpropagating error derivatives. An alternative way is to employ the weights from a trained DBN for the pre-training of an NN for classification tasks.

B. Applications

1) Speech Generation

Recently, generative models have been adopted as powerful tools for speech waveform production. WaveNet [88], [89], proposed by the DeepMind group of Google, has caught the attention of the speech synthesis community. WaveNet is a deep generative model of raw audio waveforms and is able to create a human voice which sounds very natural. It is composed of fully convolutional neural networks, involving various dilation factors. The receptive field is enlarged exponentially with the depth of the network and spans a large number of timesteps. For training, raw recordings are fed as input sequences. Synthetic utterances can be generated by sampling the trained network. Note that for producing meaningful utterances, the network's predictions are conditioned on both the audio samples and the text to be spoken. By conditioning the network on the identity of the speaker, one can use WaveNet to generate the same sentence in different voices. Interestingly, WaveNet can also be applied to model other kinds of acoustic signals, such as music. Now that WaveNet has been embedded in the Google Assistant application, it has received a lot of positive feedback [89].

As an amazing production-quality speech synthesis system, the WaveNet and its variants are embraced by many companies and research labs. Deep Voice, a truly end-to-end neural speech synthesizer employing

the WaveNet, was proposed by the Baidu Silicon Valley AI Lab in 2017 [90]–[92]. A variant of WaveNet is implemented for the audio synthesis model in this system with fewer parameters being required. In Deep Voice 2, the Tacotron [93], a text-to-speech synthesis system, was combined with a WaveNet-based spectrogram-to-audio vocoder. Evaluation results demonstrated that high-quality speech can be achieved by this integrated TTS synthesis system. A similar work describes a natural TTS synthesizer Tacotron 2 [94]. This Tacotron 2 was composed of a recurrent network, which was for mapping character embeddings to Mel-scale spectrograms, and a modified WaveNet model, which acted as a vocoder to yield time-domain waveforms from those spectrograms. This model can achieve a mean opinion score (MOS) of 4.53/5.00, which suggested very high-fidelity generated speech.

In [45], the RBM and DBN were applied to represent the distribution of the low-level spectral envelopes used for HMM-based parametric speech synthesis. Experimental results showed that both modified methods were able to generate spectral envelope parameter sequences better than the conventional Gaussian-HMM based approach. In [95], a deep autoencoder structure was proposed to extract robust spectral features for statistical parametric speech synthesis systems. By using the autoencoder, low-dimensional features can be compressed from the original high dimensional spectral envelope without degradation. A similar feature extraction procedure is also explored in [83], [96], [97].

For voice conversion, generative algorithms have been applied in many frameworks and systems to improve the naturalness, clarity and speaker individuality. In [49], a regression function constructed by a stacked joint-autoencoder was applied to a voice conversion task. Subjective listening tests were carried out to prove that the proposed approach has a higher quality and similarity than another system integrated with DNNs. In [98], a statistical voice conversion technique with the WaveNet-based waveform generation was introduced. The waveform samples of the converted voice were created by a WaveNet vocoder conditioned on the converted acoustic features. The experimental results showed that a higher conversion accuracy on speaker individuality was achieved with the proposed VC method, compared to the conventional VC techniques.

2) Other Types of Audio Signal

In [50], an autoencoder based music synthesizer was presented. The autoencoder adopted was built by a four layer deep topology and both sigmoid and ReLU activations were used. In [99] and [100], a WaveNet architecture was used to train raw audio models to

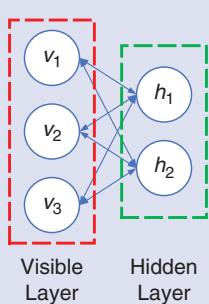


Figure 12. An example of an RBM.

produce expressive music. An automatic music generation method was proposed based on the combination of the symbolic models and the raw audio models. The musical samples achieved⁷ gave a high level of the naturalness and continuity. In another work, a powerful WaveNet-style autoencoder model was proposed to condition an autoregressive decoder for capturing longer term structure without external conditioning [101]. The model can learn a manifold of embeddings, which allows for morphing between instruments and interpolating in timbre. In [102], autoregressive discrete auto-encoders (ADAs) were explored to train autoregressive models to learn long-range correlations in raw audio waveforms. This technique can be employed to unconditionally produce piano music directly in the raw audio domain. In [103], a novel singing synthesizer was proposed based on a modified version of the WaveNet network. Compared to other works, the features produced by a parametric vocoder were modeled in this work. Given a musical score with lyrics, this approach can generate a synthetic singing voice that can learn both timbre and expression.

IV. Hybrid Model: Generative Adversarial Networks

Generative adversarial networks (GANs) are a class of neural networks developed for unsupervised machine learning. Generally, a standard GAN consists of two main components: a generator G and a discriminator D , which are trained by contesting with each other under a zero-sum game strategy [43]. The discriminator is trained by traditional supervised learning algorithms to perform two-class classification. In contrast, the generator is trained to deceive the discriminator by creating samples of the same distribution as the training data [52], [104].

These types of generative models have been employed in many applications, including high-quality image generation [105], [106], speech/image synthesis [107]–[109], im-

age translation [110], [111], semantic segmentation [112], [113] and object detection [114], [115].

A. How GANs Work

The final goal of a GAN is to learn how to generate new samples from the training samples. Assume that the probability distribution of the training data is $p(x)$. In traditional generative models, we sample from $p(x)$ to obtain new samples whereas GANs attempt to learn a mapping from a random input (or noise) to the training sample. Take image generation as an example, as shown by Fig. 14 the generator is denoted as $G(z; \theta_g)$ where θ is the network parameter and $p_z(z)$ is a prior on input noise variables. The input to the generator G is a 1-dimension random vector, z , and the output is an image produced by G . To force images that are produced to be in the same distribution as the training images, a discriminator is connected to G and denoted as $D(x; \theta_d)$. The input of D is selected randomly from either a real image or a generated

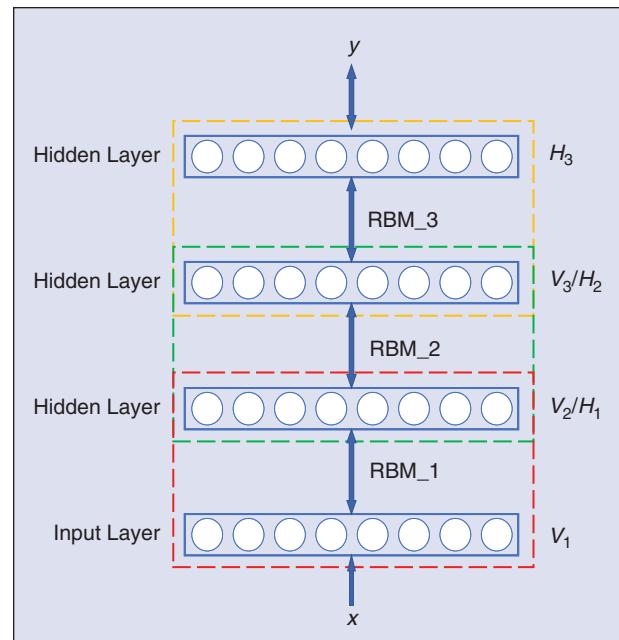


Figure 13. The network architecture of a standard DBN.

⁷<http://people.bu.edu/bkulic/projects/music/index.html>

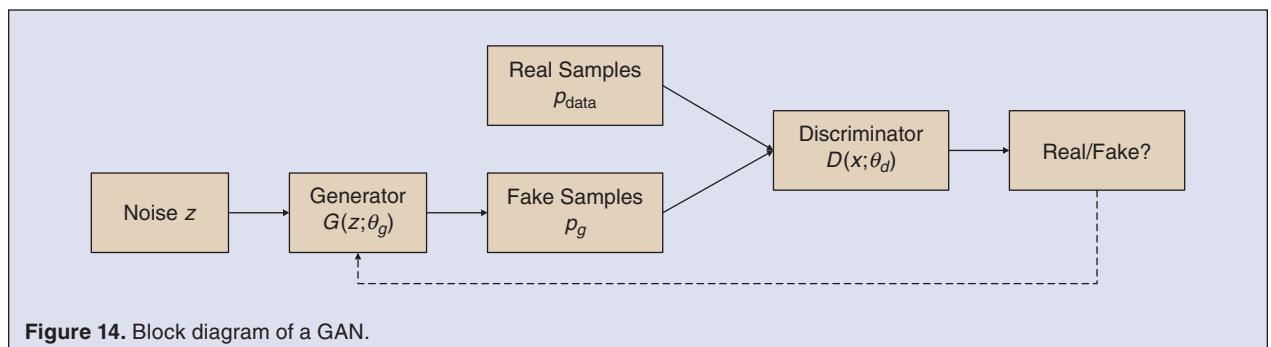


Figure 14. Block diagram of a GAN.

image from G . The output of D is a value in the range [0,1] presenting the probability of whether the input is from the real image p_{data} or generated image p_g . The feedback from D will affect the training of G to prompt the generation of images with higher fidelity, which will be fed to D for further examination. On the contrary, the discriminator is expected to distinguish ‘fake’ images from real images. This adversarial training procedure, as shown in Fig. 14, is repeated until a balance is reached between the generator G and the discriminator D .

Generally, CNNs are assigned as the primary network for the discriminator to do the image classification. The generator is, in a sense, a type of decoder: taking a vector of random noise and upsampling it to an image. The networks are each optimizing an individual but opposing objective function. As a minimax game between D and G , the value function $V(G,D)$ is given below:

$$\min_G \max_D V(D,G) = E_{x \sim p_{\text{data}}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (5)$$

In practice, rather than training G to minimize $\log(1 - D(G(z)))$, we always maximize $\log D(G(z))$ to supply sufficient gradients early in the training [43].

B. Wasserstein GAN: A Vital Improvement

Although GANs have shown great success in various applications, the training of a practicable GAN is a tricky process. This process is well known to be slow and unstable [116]. Some of the common problems are listed below:

- **Unstable training:** In the GAN’s training procedure, two adversarial models are trained simultaneously. However, the connection between the costs of these two models is independent. Updating the gradient of both models at the same time cannot guarantee a convergence. Moreover, if the performance of the discriminator is too perfect the loss function (5) will fall to zero, which results in no gradient to update. Hence it is a dilemma that the discriminator needs to behave neither too badly nor too well.

- **Mode collapse:** During the training, researchers have found that sometimes the generated images from G are in the same pattern [117]. In other words, the generator is short of sufficient diversity that the ‘fake’ images created look totally the same. This problem is referred to as mode collapse, a common failure scenario for practical applications of GANs. The generator trapped by mode collapse fails to learn the representation of the real-world data distribution and keeps creating images with low variety.

- **Lack of an evaluation metric:** The objection function of GANs cannot give a clear indication of the training progress. The relationship between the training loss curves and the training progress is confused and hard to be interpreted. There is also no explicit indicator as to the stopping criterion of the training process.

To address the above problems, many possible solutions have been discussed and analyzed and to date, the best solution is provided by the Wasserstein GAN (WGAN) [118]. In the WGAN, the Wasserstein distance is introduced to replace the conventional metrics, i.e. Kullback-Leibler (KL) or Jensen-Shannon (JS) divergence, for quantifying the similarity between two probability distributions. Compared to the KL and JS divergence, the Wasserstein distance can reflect the differential between two distributions even if there is no overlapping part. Only the Wasserstein metric keeps a smooth measure for a stable learning process using gradient descents.

The modified loss functions of the generator and the discriminator in a WGAN are $L(G) = -E_{x \sim p_g}[D(x)]$ and $L(D) = E_{x \sim p_g}[D(x)] - E_{x \sim p_{\text{data}}}[D(x)]$, respectively. According to the loss functions, the discriminator is trained to learn a K-Lipschitz continuous function to compute the Wasserstein distance, instead of directly telling the fake samples apart from the real ones. A smaller Wasserstein distance means the output of the generator is closer to the real data distribution. For maintaining the K-Lipschitz continuity of D , weights clipping is applied after updating every gradient to enforce a Lipschitz constraint.

In [119], a gradient penalty is adopted to solve the problems of exploding gradients and vanishing gradients caused by weights clipping in the original WGAN. The new model is named the WGAN-gradient penalty (WGAN-GP). The WGAN-GP has been shown to perform better than the standard WGAN and enables stable training of a wide variety of GAN models. Almost no hyper-parameter tuning is required however the time for training is significantly increased.

Some other remarkable variants of the vanilla GAN include:

- Deep Convolutional GAN (DCGAN) [105]: This is a popular CNN-based GAN, which combines the CNNs in supervised learning and the GANs in unsupervised learning. A set of constraints on the DCGAN is proposed to make this network stable to train. This architecture is a basic component in many types of GANs.
- Auxiliary Classifier GAN (AC-GAN) [108]: Although the DCGAN can produce convincing image samples, it is an intractable problem that GANs cannot

Since GANs have been widely applied for image generation and synthesis, it makes sense to consider them for speech synthesis.

generate globally coherent, high resolution samples from datasets with high variability [108]. To address this problem, label conditioning is employed in the AC-GAN that results in high resolution image samples exhibiting global coherence [104], [120], [121]. Applying a new quantitative metric proposed for image discriminability, it is shown that the samples generated from the AC-GAN are more discriminable than previous models which create lower resolution images and perform a native resize operation. The generated samples also give a comparable diversity to the training data.

- CycleGAN [122]: Analogous to language translation, image-to-image translation is defined as a problem of translating an image from one representation of a given scene to another, e.g., gray-scale to color, image to semantic labels and edge-map to photograph [122]. For image-to-image translation applications, these can be realized with the CycleGAN by learning the mapping between an input image and an output image, no matter whether the correlation information of the training samples is given or not. Comparisons against previous methods demonstrate that the CycleGAN can outperform the other approaches in quantitative experiments. The CycleGAN can be applied for image processing problems such as season transfer, collection style transfer and photo generation from paintings. A related work is [110].
- WaveGAN [123]: This is an early implementation of GANs for audio synthesis. The WaveGAN is proposed for raw audio synthesis in an unsupervised setting. Testing results suggest that WaveGAN is able to capture semantically-meaningful modes for small-vocabulary speech (such as the SC09 database analyzed in WaveGAN's work).

C. Applications in Audio Generation

Since GANs have been widely applied for image generation and synthesis, it makes sense to consider them for speech synthesis. In [123], the authors applied GANs to synthesize raw audio by introducing the WaveGAN, a time-domain approach, and the SpecGAN, a frequency-domain approach. The WaveGAN was built based on the DCGAN but modifying the transposed convolution operation to widen the receptive field for time-domain signals. Other hyper-parameters remained the same as the

DCGAN. Experiments showed that the WaveGAN can produce intelligible words and even audio from other domains like bird vocalizations or piano. The subjective evaluation demonstrated a preference for the generated samples from the WaveGAN.

In [109], an emerging topic of utilizing GANs to synthesize speech for attacking automatic speaker recognition systems was investigated. Various state-of-the-art GANs were examined by fooling a CNN-based text-independent speaker recognizers with generated Mel-spectrograms. For targeted attacks, a modified objective function was proposed to access to universal properties of speech. By applying the WGAN-GP with the modified mixed loss function, it was able to differentiate between real samples from a target speaker and real speech samples from other speakers. Resultant adversarial examples performed well for targeted and untargeted attacks to the speaker recognition system.

A CycleGAN was used in [124] with six layers of fully connected neural networks as the generator and the discriminator. The feature used for training was a mixture of a Mel-spectrogram and the first and second derivatives. A WaveNet vocoder was trained to form the speech waveform. Perceptual evaluations suggested that an effective enhancement and an improvement of the perceptual cleanliness were achieved with the help of GAN-based models. The authors also investigated the quality of the generated speech with publicly available data.

Recently, the GAN based architectures have also been explored and investigated for the music generation problem. In [125], the generator was composed using CNNs for yielding melody in the symbolic domain, while the discriminator was trained to learn the distributions of melodies. This proposed GAN, named with MidiNet, can generate melodies from scratch or by conditioning on the melody of previous bars. In another work of [126], three models were proposed for symbolic multi-track music generation under the framework called MuseGAN.⁸ These models were trained on a rock music dataset and used for piano-rolls generation (such as the bass, drums and strings). Interestingly, the models can be extended to generate additional tracks to accompany a given specific track composed by human.

Another emerging application of GANs is data augmentation for speech/acoustic signal processing. In [127],

⁸<https://salu133445.github.io/musegan/>

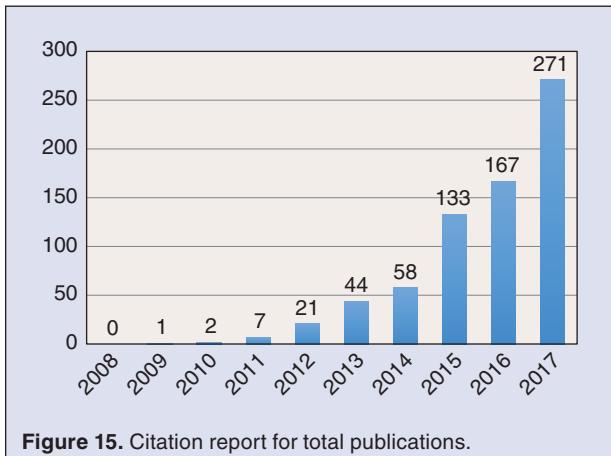


Figure 15. Citation report for total publications.

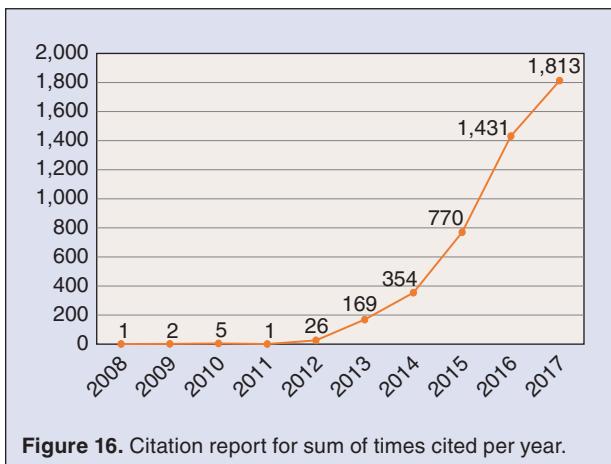


Figure 16. Citation report for sum of times cited per year.

| Table II. A comparison of deep learning based approaches on speech generation. | |
|---|-------------------------------|
| Paper | Evaluation Results |
| Zen et al. [47] | 39% (Subjective Preference) |
| Wu et al. [67] | 74% (Subjective Preference) |
| Mehri et al. [69] | 80% (Subjective Preference) |
| Fan et al. [70] | 59% (Subjective Preference) |
| Ling et al. [45] | 39% (Subjective Preference) |
| Zen et al. [64] | 3.81 ± 0.11 (5-scale MOS) |
| Oord et al. [88] | 4.21 ± 0.08 (5-scale MOS) |
| Oord et al. [89] | 4.41 ± 0.08 (5-scale MOS) |
| Ö. Arik et al. [90] | 3.94 ± 0.26 (5-scale MOS) |
| Ö. Arik et al. [91] | 3.53 ± 0.12 (5-scale MOS) |
| Ping et al. [92] | 3.78 ± 0.30 (5-scale MOS) |
| Wang et al. [93] | 3.82 ± 0.09 (5-scale MOS) |
| Shen et al. [94] | 4.53 ± 0.07 (5-scale MOS) |
| Wan et al. [96] | 4.53 ± 0.07 (5-scale MOS) |

a DCGAN-based model was used for generating more samples to augment the DCASE 2017 dataset. Random values were added to the generated samples to alleviate the issue of sample bias and over-fitting. A support vector machine (SVM) hyper-plane was trained to sift the samples of suitable quality from the generated feature pool. It was confirmed that the usage of the augmented data can improve the acoustic scene classification performance on the DCASE development set.

V. Discussion

The latest developments and applications of deep learning algorithms are reviewed in this paper. Undoubtedly, the introduction of deep generative and discriminative methods in the field of speech/acoustic signal processing boosts the accuracy and the efficiency in recognition and synthesis problems. Fig. 15 and Fig. 16 are the citation analysis from the database of the Web of Science. The search keywords are ‘speech synthesis’ and ‘deep learning.’ The bars in Fig. 15 denote the numbers of the publications recorded in the database of the Web of Science from 2008 to 2017. The curve in Fig. 16 is the total number of citations per year. It is obvious that speech synthesis applications have gripped the deep learning research community.

In Table II, evaluation results of several deep learning based speech generation methods are given for a comparison. The data listed in the table are from the best performance of the subjective experiments in each referenced paper. Given the different speech corpora used for synthesis and different implementation protocols, Table II basically demonstrates an increasing quality of the generated speech samples by employing various deep learning techniques.

In this section, the key reasons behind the success of deep learning algorithms with audio generation are discussed and several potential issues for further consideration are presented.

A. Benefits of Deep Learning

The introduction of deep learning is an important breakthrough for the speech processing community. More practical problems in audio generation can be simplified and solved in a much more refined manner. The accessibility and expandability of deep learning techniques are convenient for users to implement audio synthesis systems, not only data experts but also non-expert developers. Given enough data, computational resources and suitable algorithms, deep learning is able to address problems from a variety of aspects in acoustic signal generation.

Listed are several key advantages of deep learning based audio generation systems:

- **Effective learning of representation:** Compared to other machine learning algorithms, deep learning based techniques can learn and create more comprehensive and informative representations from raw audio signals. Deep learning based approaches applied in audio generation systems can model high-dimensional, highly correlated features efficiently. This reduces the demand for hand-crafted feature engineering, which is the most time consuming aspect for speech synthesis systems. More complex sets of features are allowed for training and evaluation in practical tasks.
- **Powerful modeling of relationships:** With the various types of activation functions, NNs have the ability to model nonlinear and complicated relationships between inputs and outputs. The unique multi-layered architecture with nonlinear operations integrates feature extraction with acoustic modeling. NNs can also model multiple simultaneous acoustic events within one frame to create waveform files with high fidelity. This is a vital property for dealing with natural signals sampled from real-world scenarios and for producing fluent conversations or background sounds.
- **Flexible setting of networks:** Unlike many other machine learning methods, specific tasks in audio generation applying deep learning can be processed with a flexible architecture with diverse combinations of DNN modules. This offers a better representation ability with increased flexibility in the parameter configuration. Additionally, there are fewer restrictions placed on the format of the inputs and allows wider commercial deployments.

B. Future Issues

Even though deep learning technology has assisted the exploitation for more audio generation applications, there are still some issues to be resolved for a more advanced realization of the “intelligible chatbot”:

- **Design of audio corpora:** With an increasing amount of sound receivers embedded in portable devices, more realistic and coarse data collected are used in the acoustic feature extraction and modeling processes. Moreover, when only a limited or an imbalanced amount of labeled audio signals is available, the performance of a deep neural network will inevitably degrade. Further efforts are required for designing elaborate audio corpora with diverse acoustic environments. More audio datasets composed of specific types of acoustic signals need to be created for recre-

ational and professional generation tasks like music composing and video post-processing.

- **Audio generation systems for practical scenarios:** With the rapid development of computational resources and deep learning algorithms, time cost on training has been effectively reduced by machines equipped with GPUs. However, it is still unworkable for most low-power portable devices like mobile phones. To relieve the computational burden, advances in hardware are necessary for the next generation intelligent equipments for both industry and academia. From the viewpoint of algorithms, several possible future directions warrant further investigation, such as more efficient models adaption for cross-lingual TTS systems and speaking style transferring. Emotional speech synthesis applying deep learning approaches is another potential application in real-world scenes.

VI. Conclusion

In this paper, the most commonly used deep generative and discriminative algorithms were reviewed. A detailed statement for the application of deep learning to speech/acoustic signal processing, especially audio generation, was provided for readers in the machine learning community. Additionally, the benefits introduced by deep learning methods and several issues of further research were discussed in detail.

Deep learning methods have shown remarkable improvement for pattern recognition and generation problems. Other machine learning approaches have been outperformed by deep learning methods due to their powerful capability of feature representation and complex model understanding. By adopting up-to-date computational equipments like GPUs, deep learning techniques demonstrate impressive performance for yielding human-like speech. It is indeed the case that there is still a long way to go before the realization of a general intelligence or strong AI. However, there is wide confidence that deep learning will receive more enthusiasm and motivation for broader applications in the future.

Acknowledgment

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.



Yuanjun Zhao received the Bachelor degree and Master degree in Electronic and Communication Engineering from the Harbin Institute of Technology (HIT) in 2013 and 2015, respectively. Now he is a Ph.D. candidate in the School of Electrical,

Electronic and Computer Engineering from The University of Western Australia (UWA) since 2016. His research interests include speech signal processing, automatic speaker recognition and speaker spoofing detection.



Xianjun Xia received the Bachelor degree in Electronic Engineering from the Hefei University of Technology (HFUT) in 2011 and Master degree in Signal and Information Processing from the University of Science and Technology of China (USTC) in 2014. Now he is a Ph.D. candidate in the School of Electrical, Electronic and Computer Engineering from The University of Western Australia (UWA) since 2015. His research interests include acoustic event detection, machine learning, speech and audio signal processing.



Roberto Tognoni (M89-SM04) received the Ph.D degree in 1989 from the University of Western Australia. He joined the School of Electrical, Electronic and Computer Engineering at The University of Western Australia in 1988, where he is now currently an Associate Professor. He leads the Signal Processing and Recognition Lab and his research activities in signal processing and pattern recognition include: feature extraction and enhancement of audio signals, statistical and neural network models for speech and speaker recognition, audio-visual recognition and biometrics, and related aspects of language modelling and understanding. He has published over 150 refereed journal and conference papers in the areas of signal processing and recognition, the chief investigator on three Australian Research Council Discovery Project research grants since 2009, was an Associate Editor for *IEEE Signal Processing Magazine Lecture Notes* and *IEEE Transactions on Speech, Audio and Language Processing* from 2012 to 2016 and is currently the Area Editor for the *IEEE Signal Processing Magazine*, Columns and Forums.

References

- [1] T. A. Edison, "The phonograph and its future," *North Amer. Rev.*, vol. 126, no. 262, pp. 527–536, 1878.
- [2] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.
- [3] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Commun.*, vol. 51, no. 11, pp. 1039–1064, 2009.
- [4] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *1996 IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Conf. Proc. (ICASSP-96.)*, vol. 1, pp. 373–376.
- [5] M. Schröder, M. Charfuelan, S. Pammi, and I. Steiner, "Open source voice creation toolkit for the MARY TTS platform," in *Proc. 12th Annu. Conf. Int. Speech Communication Association*, 2011.
- [6] J. Yamagishi, "An introduction to HMM-based speech synthesis," National Institute of Informatics, Tech. Rep., 2006.
- [7] A. F. Machado and M. Queiroz, "Voice conversion: A critical survey," in *Proc. Sound and Music Computing (SMC)*, 2010, pp. 1–8.
- [8] D. Childers, B. Yegnanarayana, and K. Wu, "Voice conversion: Factors responsible for quality," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, (ICASSP'85)*, vol. 10. 1985, pp. 748–751.
- [9] T. Toda, H. Saruwatari, and K. Shikano, "Voice conversion algorithm based on Gaussian mixture model with dynamic frequency warping of straight spectrum," in *Proc. 2001 IEEE Int. Conf. Acoustics, Speech, and Signal Processing, (ICASSP'01)*, vol. 2. pp. 841–844.
- [10] T. Toda et al., "The voice conversion challenge 2016," in *Proc. Inter-speech*, 2016, pp. 1632–1636.
- [11] Y. Tabet and M. Boughazi, "Speech synthesis techniques: A survey," in *Proc. 2011 7th Int. Workshop Systems, Signal Processing and their Applications (WOSSPA)*. IEEE, pp. 67–70.
- [12] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Commun.*, vol. 66, pp. 130–153, 2015.
- [13] M. Todisco, H. Delgado, and N. Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Comput. Speech Lang.*, vol. 45, pp. 516–535, 2017.
- [14] Y. Zhao, R. Tognoni, and V. Seeram, "Compressed high dimensional features for speaker spoofing detection," in *Proc. Asia-Pacific Signal and Information Processing Association Annu. Summit and Conf. (APSIPA ASC)*. IEEE, 2017, pp. 569–572.
- [15] C. Zhang, C. Yu, and J. H. Hansen, "An investigation of deep-learning frameworks for speaker verification antispoofing," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 4, pp. 684–694, 2017.
- [16] N. Agarwala, Y. Inoue, and A. Sly, "Music composition using recurrent neural networks," Technical Report, Tech. Rep., 2017.
- [17] J.-P. Briot, G. Hadjeres, and F. Pachet, "Deep learning techniques for music generation-a survey," arXiv Preprint, arXiv:1709.01620, 2017.
- [18] A. Huang and R. Wu, "Deep learning for music," arXiv Preprint, arXiv:1606.04930, 2016.
- [19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [20] K. Sundararajan and D. L. Woodard, "Deep learning for biometrics: A survey," *ACM Comput. Surv. (CSUR)*, vol. 51, no. 3, p. 65, 2018.
- [21] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.
- [22] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Rev.*, vol. 65, no. 6, p. 386, 1958.
- [23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ., San Diego, La Jolla Inst. for Cognitive Science, Tech. Rep., 1985.
- [24] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015.
- [25] P. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," 1974.
- [26] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [27] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, no. Feb, pp. 625–660, 2010.
- [28] C. Poulton, S. Chopra, Y. L. Cun et al., "Efficient learning of sparse representations with an energy-based model," in *Proc. Advances in Neural Information Processing Systems*, 2007, pp. 1137–1144.
- [29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [30] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- [31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
- [32] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. European Conf. Computer Vision*. Springer-Verlag, 2014, pp. 818–833.
- [33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv Preprint, arXiv:1409.1556, 2014.
- [34] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [36] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [37] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Machine Learning*. ACM, 2008, pp. 1096–1103.
- [38] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. Dec, pp. 3371–3408, 2010.
- [39] A. Makhzani and B. Frey, "K-sparse autoencoders," arXiv Preprint, arXiv:1312.5663, 2013.
- [40] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," arXiv Preprint, arXiv:1312.6114, 2013.
- [41] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. 28th Int. Conf. Machine Learning*. Omnipress, 2011, pp. 833–840.
- [42] L. Deng and N. Jaitly, "Deep discriminative and generative models for speech pattern recognition," in *Handbook of Pattern Recognition and Computer Vision*. New York: World Scientific, 2016, pp. 27–52.
- [43] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
- [44] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations Trends® Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2014.
- [45] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2129–2139, 2013.
- [46] S. Kang, X. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proc. 2013 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8012–8016.
- [47] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. 2013 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7962–7966.
- [48] L.-J. Liu, L.-H. Chen, Z.-H. Ling, and L.-R. Dai, "Using bidirectional associative memories for joint spectral envelope modeling in voice conversion," in *Proc. 2014 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7884–7888.
- [49] S. H. Mohammadi and A. Kain, "A voice conversion mapping function based on a stacked joint-autoencoder," in *Proc. INTERSPEECH*, 2016, pp. 1647–1651.
- [50] J. Colonel, C. Curro, and S. Keene, "Improving neural net auto encoders for music synthesis," in *Audio Engineering Society Convention 143*. Audio Engineering Society, 2017.
- [51] A. M. Sarroff and M. A. Casey, "Musical audio synthesis using auto-encoding neural nets," in *Proc. ICMC*, 2014.
- [52] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA: MIT Press, 2016.
- [53] J. Gu et al., "Recent advances in convolutional neural networks," *Pattern Recog.*, vol. 77, pp. 354–377, 2018.
- [54] M. Lin, Q. Chen, and S. Yan, "Network in network," arXiv Preprint, arXiv:1312.4400, 2013.
- [55] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," arXiv Preprint, arXiv:1506.00019, 2015.
- [56] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artificial Intelligence and Statistics*, 2011, pp. 315–323.
- [57] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Machine Learning (ICML-10)*, 2010, pp. 807–814.
- [58] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [59] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," 1999.
- [60] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv Preprint, arXiv:1406.1078, 2014.
- [61] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv Preprint, arXiv:1412.3555, 2014.
- [62] R. Dey and F. M. Salemt, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. 2017 IEEE 60th Int. Midwest Symp. Circuits and Systems (MWSCAS)*, pp. 1597–1600.
- [63] Z.-H. Ling et al., "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 35–52, 2015.
- [64] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proc. 2014 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3844–3848.
- [65] J. Lorenzo-Trueba, G. E. Henter, S. Takaki, J. Yamagishi, Y. Morino, and Y. Ochiai, "Investigating different representations for modeling and controlling multiple emotions in DNN-based speech synthesis," *Speech Commun.*, vol. 99, pp. 135–143, 2018.
- [66] J. Chorowski, R. J. Weiss, R. A. Saurous, and S. Bengio, "On using backpropagation for speech texture generation and voice conversion," in *Proc. 2018 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 2256–2260.
- [67] Z. Wu and S. King, "Investigating gated recurrent networks for speech synthesis," in *Proc. 2016 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 5140–5144.
- [68] B. Li and H. Zen, "Multi-language multi-speaker acoustic modeling for LSTM-RNN based statistical parametric speech synthesis," in *Proc. INTERSPEECH*, 2016, pp. 2468–2472.
- [69] S. Mehri et al., "SampleRNN: An unconditional end-to-end neural audio generation model," arXiv Preprint, arXiv:1612.07837, 2016.
- [70] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. 15th Annu. Conf. Int. Speech Communication Association*, 2014.
- [71] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bi-directional long short-term memory based recurrent neural networks," in *Proc. 2015 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4869–4873.
- [72] H. Zhu et al., "Xiaoice Band: A melody and arrangement generation framework for pop music," in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*. ACM, 2018, pp. 2837–2846.
- [73] D. D. Johnson, "Generating polyphonic music using tied parallel networks," in *Proc. Int. Conf. Evolutionary and Biologically Inspired Music and Art*. Springer-Verlag, 2017, pp. 128–143.
- [74] G. Brunner, Y. Wang, R. Wattenhofer, and J. Wiesendanger, "JamBot: Music theory aware chord based generation of polyphonic music with LSTMs," in *Proc. 2017 IEEE 29th Int. Conf. Tools with Artificial Intelligence (ICTAI)*, pp. 519–526.
- [75] M. Blaauw and J. Bonada, "A singing synthesizer based on pixel-CNN," in *María de Maeztu Seminar on Music Knowledge Extraction Using Machine Learning (Collocated with NIPS)*. 2016. Accessed on: Oct. 1, 2017. [Online]. Available: http://www.dtic.upf.edu/~mblaauw/MdM_NIPS_seminar/
- [76] A. van den Oord et al., "Conditional image generation with pixel-CNN decoders," in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 4790–4798.
- [77] M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on deep neural networks," in *Proc. INTERSPEECH*, 2016, pp. 2478–2482.
- [78] E. Gómez, M. Blaauw, J. Bonada, P. Chandna, and H. Cuesta, "Deep learning for singing processing: Achievements, challenges and impact on singers and listeners," arXiv Preprint, arXiv:1807.03046, 2018.
- [79] D. Makris, M. Kaliakatos-Papakostas, I. Karydis, and K. L. Keranidis, "Combining LSTM and feed forward neural networks for conditional rhythm composition," in *Proc. Int. Conf. Engineering Applications of Neural Networks*. Springer-Verlag, 2017, pp. 570–582.
- [80] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [81] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. 2012 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4153–4156.
- [82] K. Janod et al., "Denoised bottleneck features from deep autoencoders for telephone conversation analysis," *IEEE/ACM Trans. Audio, Speech Lang. Process. (TASLP)*, vol. 25, no. 9, pp. 1809–1820, 2017.
- [83] S. Takaki and J. Yamagishi, "A deep auto-encoder based low-dimensional feature extraction from FFT spectral envelopes for statistical parametric speech synthesis," in *Proc. 2016 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5535–5539.

- [84] Y. Pu et al., "Variational autoencoder for deep learning of images, labels and captions," in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 2352–2360.
- [85] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *Proc. European Conf. Computer Vision*. Springer-Verlag, 2016, pp. 835–851.
- [86] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 3483–3491.
- [87] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*. New York: Springer-Verlag, 2012, pp. 599–619.
- [88] A. van den Oord et al., "WaveNet: A generative model for raw audio," in *Proc. 9th ISCA Speech Synthesis Workshop*, Sunnyvale, CA, Sept. 13–15, 2016, p. 125.
- [89] A. v d Oord et al., "Parallel WaveNet: Fast high-fidelity speech synthesis," arXiv Preprint, arXiv:1711.10433, 2017.
- [90] S. Ö. Arik et al., "Deep voice: Real-time neural text-to-speech," in *Proc. 34th Int. Conf. Machine Learning, Machine Learning Research*, vol. 70. D. Precup and Y. W. Teh, Eds. Sydney, Australia: PMLR, International Convention Centre, Aug. 6–11, 2017, pp. 195–204.
- [91] A. Gibiansky et al., "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in Neural Information Processing Systems 30*, I. Guyon et al., Eds. Curran Associates, 2017, pp. 2962–2970.
- [92] W. Ping et al., "Deep voice 3: 2000-speaker neural text-to-speech," arXiv Preprint, arXiv:1710.07654, 2017.
- [93] Y. Wang et al., "Tacotron: Towards end-to-end speech synthesis," in *Proc. INTERSPEECH*, 2017, pp. 4006–4010.
- [94] J. Shen et al., "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *Proc. 2018 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 4779–4783.
- [95] H. Choi, J. Kim, J. Park, J. Kim, and M. Hahn, "Low-dimensional representation of spectral envelope using deep auto-encoder for speech synthesis," in *Proc. 2018 2nd Int. Conf. Mechatronics Systems and Control Engineering*. ACM, pp. 107–111.
- [96] V. Wan, Y. Agiomyrgiannakis, H. Silen, and J. Vt, "Google's next-generation real-time unit-selection synthesizer using sequence-to-sequence LSTM-based autoencoders," in *Proc. Interspeech 2017*, pp. 1143–1147.
- [97] Y.-J. Hu and Z.-H. Ling, "Extracting spectral features using deep autoencoders with binary distributed hidden units for statistical parametric speech synthesis," *IEEE/ACM Trans. Audio, Speech Lang. Process. (TASLP)*, vol. 26, no. 4, pp. 713–724, 2018.
- [98] K. Kobayashi, T. Hayashi, A. Tamamori, and T. Toda, "Statistical voice conversion with WaveNet-based waveform generation," in *Proc. Interspeech*, vol. 2017. 2017, pp. 1138–1142.
- [99] R. Manzelli, V. Thakkar, A. Siahkamari, and B. Kulic, "An end to end model for automatic music generation: Combining deep raw and symbolic audio networks," in *Proc. Musical Metacreation Workshop at 9th Int. Conf. Computational Creativity*, Salamanca, Spain, 2018.
- [100] R. Manzelli, V. Thakkar, A. Siahkamari, and B. Kulic, "Conditioning deep generative raw audio models for structured automatic music," arXiv Preprint, arXiv:1806.09905, 2018.
- [101] J. Engel et al., "Neural audio synthesis of musical notes with WaveNet autoencoders," in *Proc. 34th Int. Conf. Machine Learning, Machine Learning Research*, vol. 70. D. Precup and Y. W. Teh, Eds. PMLR, Aug. 6–11 2017, pp. 1068–1077.
- [102] S. Dieleman, A. v d Oord, and K. Simonyan, "The challenge of realistic music generation: Modelling raw audio at scale," arXiv Preprint, arXiv:1806.10474, 2018.
- [103] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," *Appl. Sci.*, vol. 7, no. 12, p. 1313, 2017.
- [104] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.
- [105] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv Preprint, arXiv:1511.06434, 2015.
- [106] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," *CVPR*, vol. 2, no. 5, p. 7, 2017.
- [107] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proc. 33rd Int. Conf. Machine Learning, Machine Learning Research*, vol. 48. M. F. Balcan and K. Q. Weinberger, Eds. New York: PMLR, June 20–22, 2016, pp. 1060–1069.
- [108] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Machine Learning, Machine Learning Research*, vol. 70. D. Precup and Y. W. Teh, Eds. Sydney, Australia: PMLR, International Convention Centre, Aug. 6–11 2017, pp. 2642–2651.
- [109] W. Cai, A. Doshi, and R. Valle, "Attacking speaker recognition with deep generative models," arXiv Preprint, arXiv:1801.02384, 2018.
- [110] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [111] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 469–477.
- [112] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," in *Proc. NIPS Workshop Adversarial Training*, 2016.
- [113] W. Zhu, X. Xiang, T. D. Tran, and X. Xie, "Adversarial deep structural networks for mammographic mass segmentation," arXiv Preprint, arXiv:1612.05970, 2016.
- [114] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-RCNN: Hard positive generation via adversary for object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [115] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [116] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," arXiv Preprint, arXiv:1701.04862, 2017.
- [117] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," arXiv Preprint, arXiv:1611.02163, 2016.
- [118] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Machine Learning, Machine Learning Research*, vol. 70. D. Precup and Y. W. Teh, Eds. Sydney, Australia: PMLR, International Convention Centre, Aug. 6–11, 2017, pp. 214–223.
- [119] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.
- [120] A. Odena, "Semi-supervised learning with generative adversarial networks," arXiv Preprint, arXiv:1606.01583, 2016.
- [121] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 3387–3395.
- [122] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [123] C. Donahue, J. McAuley, and M. Puckette, "Synthesizing audio with generative adversarial networks," arXiv Preprint, arXiv:1802.04208, 2018.
- [124] J. Lorenzo-Trueba, F. Fang, X. Wang, I. Echizen, J. Yamagishi, and T. Kinnunen, "Can we steal your vocal identity from the internet? Initial investigation of cloning Obamas voice using GAN, WaveNet and low-quality found data," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pp. 240–247.
- [125] L. Yang, S. Chou, and Y. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proc. 18th Int. Society for Music Information Retrieval Conf., ISMIR 2017*, Suzhou, China, Oct. 23–27, 2017, pp. 324–331.
- [126] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," 2018. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17286>
- [127] S. Mun, S. Park, D. K. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane," in *Proc. DCASE*, 2017, pp. 93–97.
- [128] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 3856–3866.

Recent Development in Public Transport Network Analysis From the Complex Network Perspective



Tanuja Shanmukhappa, Ivan Wang-Hei Ho, Chi K. Tse, and Kin K. Leung

©ISTOCKPHOTO.COM/MUCHMANIA

Abstract

A graph, comprising a set of nodes connected by edges, is one of the simplest yet remarkably useful mathematical structures for the analysis of real-world complex systems. Network theory, being an application-based extension of graph theory, has been applied to a wide variety of real-world systems involving complex interconnection of subsystems. The application of network theory has permitted in-depth understanding of connectivity, topologies, and operations of many practical networked systems as well as the roles that various parameters play in determining the performance of such systems. In the field of transportation networks, however, the use of graph theory has been relatively much less explored, and this motivates us to bring together the recent development in the field

of public transport analysis from a graph theoretic perspective. In this paper, we focus on ground transportation, and in particular the *bus transport network* (BTN) and *metro transport network* (MTN), since the two types of networks are widely used by the public and their performances have significant impact to people's life. In the course of our analysis, various network parameters are introduced to probe into the impact of topologies and their relative merits and demerits in transportation. The various local and global properties evaluated as part of the topological analysis provide a common platform to comprehend and decipher the inherent network features that are partly encoded in their topological properties. Overall, this paper gives a detailed exposition of recent development in the use of graph theory in public transport network analysis, and summarizes the key results that offer important insights for government agencies and public transport system operators to plan, design, and optimize future public transport networks in order to achieve more efficient and robust services.

Digital Object Identifier 10.1109/MCAS.2019.2945211

Date of current version: 19 November 2019

I. Introduction

Public transportation systems form a vital part of our infrastructure that permits massive flow of commuters within a city and between cities. In order to meet the rising standards of living of the society, transportation networks have to keep abreast of the need of commuters with respect to the ever increasing demand of reducing the traveling time and extending the area covered. At the same time, transportation networks are facing series of challenges, including satisfying the ever increasing passenger volume, achieving long-term sustainability, and improving the quality of service. Such challenges are encountered at various levels of operation, ranging from infrastructure deployment to optimal route planning, and the problems are addressed from different angles depending on the discipline of study such as urban planning, regional science, geography, engineering, etc. The literature abounds with diverse methodologies adopted in various disciplines to represent, perceive and analyze the complex dynamics of public transport systems, among which, Geographic Information System (GIS), graph theory, mathematical programming, and agent-based modeling are most commonly adopted [1].

Motivated by the notable contributions of network theory [2], application of graph theoretic concepts in the analysis of public transport networks (PTN) has attracted significant attention, and today, it is one of the most widely employed approaches to understand the nature of connectivity in PTNs. The representation of a PTN as a complex network, together with the adoption of some concepts from statistical physics, offers remarkable advantages in the modeling and analysis of nonlinear and dynamic PTN structures. Specifically, the analysis of PTNs using network theory permits the use of a common platform on which to comprehend and decipher the inherent network features that are encoded in the topological properties. Moreover, to apply the concepts of complex networks, one should understand the language of graph theory, as a prerequisite, where a network is typically represented as a graph consisting of a set of *nodes* interconnected by a set of *edges*.

Graph theory and network theory, despite being rooted historically in mathematics, has found applications in statistical physics, biology, social sciences, finance, and engineering. One of the oldest instances of using the notion of graph theory to analyze a real-world problem dates back to the 17th Century when Leonhard Euler used the concept of nodes and edges to solve the problem of seven bridges of Königsberg, a notable problem in the history of mathematics [3]. However, notable us-

age of graph theory was found by Gustav Kirchhoff who employed nodes and edges to calculate voltages and currents in electric circuits, nowadays widely known as Kirchhoff's laws [4]. Subsequently, many real-world networks were analyzed using graph theory with significant contributions from the fields of social networks (world wide web) and biological networks, and later from other fields including friendship networks, relationship in social media, food web, metabolism, professional ties, author and co-author relations, citation networks, computer virus flow, network router analysis, chemical reactions, neural networks, transportation networks, etc. From the literature, it was evident that modeling various large real-world network structures as graphs, and analyzing their behavior from a network perspective, facilitated better understanding of both the global and local properties of the network. Thus, this domain of study has attracted a humongous amount of research interest in the past two decades [5]–[7]. Although a lot of real-world complex systems have been analyzed using graph theory, little attention has been paid to the field of transportation networks which is an active research area among researchers in transportation and logistics.

Although a public transport network can either be unimodal or multi-modal, we focus on two major types of public transportation, namely, the bus transport network (BTN) and metro transport network (MTN), since we believe that these two types of networks are most widely used by the public to meet their daily commuting needs. In this article, the recent contributions and the concepts employed in the topological analysis of public transport networks are discussed. Our focus is on the understanding of various network parameters and approaches employed to analyze the topology of a PTN [5]–[7]. Moreover, a brief discussion on the fundamental graph theoretic concepts will be made whenever necessary.

The remainder of the paper is structured as follows. Section II introduces a few preliminary steps to be followed to construct a real-world network topology from given datasets, i.e., collecting the real-world datasets from various online sources, and data mining to extract useful information from both computational and visualization perspectives. Section III presents various spaces of graph representation for studying the topological representation of PTNs. Section IV discusses in detail the contributions of previous works in terms of the use of appropriate network parameters that aid PTN analysis. Section V focuses on some distinctive contributions accomplished in PTN analysis which might pave the way for future research or some food for thought. Finally, in

T. Shanmukhappa, I. W.-H. Ho, and C. K. Tse are with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. K. K. Leung is with the Department of Electrical and Electronic Engineering, and Computing, Imperial College London, United Kingdom.

Section VI, a few important conclusions are drawn, and the possible scope of future work is discussed.

II. Data Collection, Mining and Visualization

Although significant research interest in the field of network science theory has been cultivated for several decades, applying the established concepts to real-world data has been practiced only recently as a consequence of the availability of real-world datasets and the high-end tools for processing such huge datasets. With the aid of real-world datasets, a network topology which closely mimics the real-world structure is generated using the concepts of graph theory. Building a network topology forms the fundamental and important aspect in a PTN analysis since the course of the defined topology significantly influences the understanding of both the local and global aspects of a network. A list of online sources and relevant datasets are given in Appendix A. The extracted datasets include information on

- i) List of *stops/stations* along with their id's, names, latitude/northing and longitude/easting data.

- ii) List of *routes/sequence-of-stops* along with their stop sequence id's and names for the inbound, outbound, and round-trip routes.

Here, a *stop* or *station* is a designated place allocated to pick up or drop off passengers, and a *route* (sequence of stops) is a path taken to reach the destination from a source along the intermediate stops. Furthermore, other information such as the list of routes operated by different operators, end-to-end travel cost, frequency of operation, specific day and time (e.g., weekday, weekend, special days, peak hours, off-peak hours, day-time, night-time routes), etc., are also available in a few datasets.

Like other complex networks, the availability of huge data has posed big challenge to transport network analysis. Fortunately, the obtained datasets for PTNs are relatively midget, and can be processed in a reasonable time. Here, we describe three basic steps in mining the crude datasets for extracting meaningful information:

Step 1: Eliminate the anomalies that are commonly observed in the extracted datasets, e.g., data redundancies

Appendix A: Online Sources

| City | Country | Source | Ref. |
|--------------------------------|-----------|--|------|
| Bus Transport Network | | | |
| Hangzhou | China | www.hzbuda.com.cn | [23] |
| Chennai | India | www.mtcbus.org/ | [11] |
| Ahmedabad | India | www.ahmedabadbrts.org/web/commuters.html | [11] |
| Delhi | India | delhitravelhelp.in/StopsOfBus.aspx | [11] |
| Hyderabad | India | www.hyderabadbusroutes.com | [11] |
| Kolkata | India | www.kolkataonline.in | [11] |
| Mumbai | India | github.com/transitmetrics/ntd/tree/master | [11] |
| - | Singapore | www.streetdirectory.com.sg/ | - |
| Hong Kong | China | data.gov.hk/en-data/category/transport?organization= hk-td/ | [68] |
| London | UK | data.london.gov.uk/dataset/tfl-bus-stop-locations-and-routes | - |
| Bengaluru | India | opencity.in/topic/transportation/ | - |
| - | Australia | opendata.transport.nsw.gov.au/search/type/dataset | - |
| - | - | www.apta.com | [17] |
| Metro Transport Network | | | |
| Beijing | China | www.ebeijing.gov.cn/feature_2/BeijingSubway/ | [33] |
| Shanghai | China | service.shmetro.com/en/ | [31] |
| Hong Kong | China | www.mtr.com.hk/en/customer/tourist/index.php | [32] |
| Tokyo | Japan | www.tokyometro.jp/en/subwaymap/index.html | [32] |
| London | UK | tfl.gov.uk/maps/track?intcmp=40400 | [32] |
| New York | America | web.mta.info/maps/submap.html | [32] |
| Boston | America | mbta.com/schedules/subway | [69] |
| Paris | France | parisbytrain.com/paris-metro/ | [32] |
| Seoul | Korea | www.korea4expats.com/korean-subways.php | [35] |

with respect to the locations of public transport stops or routes, missing information in the sequence of stops along a route, allocation of multiple id's to a specific stop or route, missing information on the geographical location of a few stops, etc.

Step 2: Process the crude data obtained in Step 1 to permit further analysis. This involves the following procedure:

- Since PTNs belong to the category of spatial networks, understanding the topological behavior along with spatial information will facilitate better network analysis. The spatial information of public transport stops listed in the datasets are either easting-northing or latitude-longitude. However, since many of the network visualization tools adopts latitude and longitude information for displaying the spatial locations of the stations, it is useful to convert easting and northing data to latitude and longitude using tools like ArcGIS [8]. Before the conversion, a suitable global coordinate system (e.g., WGS84) should be chosen based on the information about local coordinate systems (e.g., OSGB36 for London and HK1980 for Hong Kong) provided by the local survey departments [9].
- In some datasets, the numbers assigned for the stops are typically non-sequential in nature, posing computational challenges to analysis, e.g., in generating adjacency matrix as discussed in Step 3. Thus, it is necessary to map the list of id's (both routes

and stops) extracted from the database with sequentially mapped numbers. This mapping of original stop id's with sequentially mapped id's makes it less arduous to further process the data.

- The concept of short distance station pairs (SSPs) has been commonly adopted to represent a group of stations as a single (merged) station [10]–[12]. Assigning new id's to such SSPs according to the sequential mapping carried out in Step 2 is recommended to facilitate identification of SSPs in a network. The clustering of multiple stations into one station can be based on geographical closeness, similar names for nearby stations, stations within a specific walkable catchment, etc. Although different terminologies have been used, the essential idea of SSPs has been reported in several sources [10], [11], [13]. The idea behind identifying SSPs is to establish a virtual connectivity among the nodes, especially when a large number of SSPs are observed in the network [10], [13]. However, as discussed in ref. [13], when combining multiple nodes as a single node based on their geographical closeness, the actual definition of geographical closeness is always a matter of choice. A distance threshold (d_{th}) is needed to define the closeness of two nodes and can be chosen judiciously by observing the distribution pattern of geographical distances between successive stations (d_{ij}) in a network. However, it should

Table I.
Graph type and space of representation used in various PTN analyses.

| | Directed | Undirected | Weighted | Unweighted | References |
|--------------------------------|----------|------------|----------|------------|-------------------------------|
| Bus Transport Network | | | | | |
| L-space | ✓ | • | • | ✓ | [15] [16] [17] |
| | ✓ | • | ✓ | • | [11] [18] [12] [19] [20] [21] |
| P-space | ✓ | • | • | ✓ | [10] [13] [22] |
| | ✓ | • | ✓ | • | [16] [17] [23] [18] [24] [25] |
| C-space | ✓ | • | • | ✓ | [10] [20] [22] [26] |
| | • | ✓ | • | ✓ | [17] [27] |
| Metro Transport Network | | | | | |
| L-space | • | ✓ | • | ✓ | [28] [29] |
| | • | ✓ | ✓ | • | [30] [31] [32] [33] [34] [35] |

be noted that choosing an extremely small value of d_{th} creates a lot of SSPs in a dense network, whereas a large value of d_{th} is meaningless, since a long walking distance to reach another station in the network is unreasonable. In either of the cases, the chosen value of d_{th} may bias the understanding of network behavior [13]. Hence, a careful selection of the d_{th} is important. SSPs are more prevalently observed in bus transport networks as compared to metro transport networks.

Step 3: Generate the topology of a PTN from the data extracted in Step 2. Initially, based on the graph type and the space of representation, a square adjacency matrix A with dimension $N \times N$ and elements a_{ij} can be derived to describe the connection between node pair n_i and n_j . The element $a_{ij} = 1$ if there exists a connection between nodes n_i and n_j , and 0 otherwise. A graph can either be directed (digraph), undirected, weighted or unweighted. The intent of choosing the graph type solely depends on the necessity of the type of analysis to be accomplished. For the analysis of transport structures, especially bus transport structures, a directed graph is often chosen since the inbound and outbound routes have different travel paths servicing different stations (except the round-trip journey routes). However, an undirected graph is typically chosen in the analysis of metro transport networks where the inbound and outbound travel paths remain the same for a vast majority of routes. Furthermore, depending on the aim of the network analysis, the graph can be represented in various spaces of representation as will be discussed in Section III. Thus, the type of graphs (directed, undirected, weighted, unweighted) along with the space of representation (L-, P-, B- and C-space) defines the topology of a

PTN structure to be examined. Table I shows the graph type and the space of representation chosen in various PTN analysis in the literature.

Finally, for visualizing a network, there are many open source network visualization tools, and the selection would depend on the need of the analysis. For a comparison of different visualization tools, interested readers are referred to ref. [14].

III. Spaces of Network Representation

In this section, we describe different spaces of network representation together with the adjacency matrix representation for analysis of public transport networks. Our discussion will follow the basics introduced in Kurant and Thiran [15] and Ferber *et al.* [17] for representing a public transport network in different spaces of network representation, as shown in Fig. 1. The various topological representations are fundamentally related to how the network and its parameters are being perceived. For instance, different aspects of interest may include information about the stations having more routes traversing through them, the most significant station in a network in terms of connectivity, the routes servicing more stations, edges with more overlapped routes, the number of transfers needed to reach two different stations in a network, and so on. Fig. 1 shows the most commonly used representations of a PTN analysis along with their adjacency matrix entries.

- i) A graph in L-space, also called the space-of-stations, is shown in Fig. 1(b). In an L-space graph, a public transport stop is treated as a node, and a pair of nodes are connected by an edge if there is at least one route servicing the two stops consecutively. The L-space representation is the most

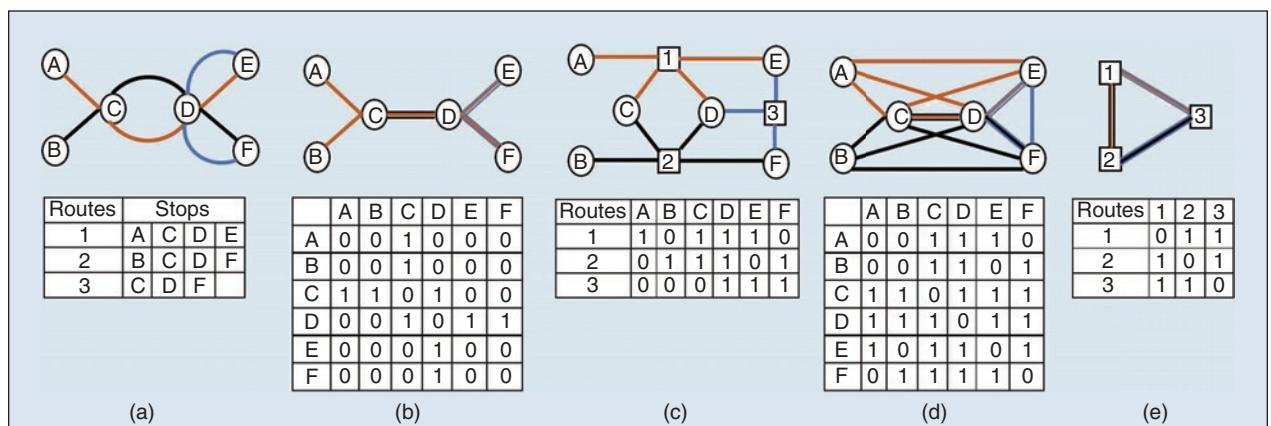


Figure 1. (a) Simple public transport map with stations A–F being serviced by route no. 1 (shaded orange), no. 2 (black), and no. 3 (blue); (b) L-space graph; (c) B-space bipartite graph (route nodes are shown as squares); (d) P-space graph (complete sub-graph corresponding to route no. 1 is highlighted in orange); (e) C-space graph of routes. The matrix of connectivity is shown below the corresponding network representation.

extensively used representation in the analysis of PTNs since it signifies the actual physical infrastructure that exists in a real-world network, and renders useful information on relationship between the nodes.

- ii) A graph in B-space, also called a bipartite graph, is shown in Fig. 1(c), where both the routes and stops are represented by nodes. A route node is connected to all the stops it services, and a stop node is connected to all the routes servicing it. There is no directed edge between nodes of the same type, i.e., an edge will not exist between two route nodes or stop nodes. A graph in the B-space will be undirected. Although analysis of PTNs using bipartite graphs finds limited application, the one mode projection of a bipartite graph into the P-space (node projected) and the C-space (route projected) has gained significant attention.
- iii) A graph in P-space is also called space-of-changes, space-of-transfers, or stop-unipartite graph, and is shown in Fig. 1(d). In the P-space, the stops are represented by nodes and every possible pair of nodes that can be reached without making any transfers are linked by edges (stops serviced by a single route). A graph in the P-space can be undirected or directed depending on the type of transport networks (BTN or MTN) under study. The P-space representation renders useful information for studying the transfers between different routes since the neighbors of a node in the P-space representation are the set of nodes that can be reached with or without making a transfer. Hence, the node set associated with a specific route forms a clique or a complete subgraph.
- iv) A graph in the C-space is also called route-unipartite graph, as shown in Fig. 1(e). In the C-space, the nodes are the routes and two nodes are connected by an edge if they service a common set of stop(s) along their journeys. A graph in the C-space can be directed or undirected depending on the type of networks under study (BTN or MTN).

Table II shows the allowed graph types (directed or undirected) with respect to various spaces of network

Table II.
Allowed graph type under various spaces of representation.

| Space | Directed | Undirected |
|-------|----------|------------|
| L | Yes | Yes |
| B | No | Yes |
| P | No | Yes |
| C | Yes | Yes |

representation (L-, B-, P- and C-space) and the type of transport networks (bus or metro).

IV. Overview of Topological Analysis of Public Transport Networks

Network Science by itself has no strong association with any single field of study as its applications can be found in a great variety of real-world systems. There are a handful of parameters commonly used for analyzing complex networks. In this section, some key network parameters that aid the understanding of public transport networks are discussed. For brevity and convenience of discussion, a nomenclature list is given in Table III.

The topology of the network under analysis is represented as a graph G , which is an ordered pair comprising a set of nodes (V) and a set of edges (E), i.e., $G = (V, E)$ such that

$$V = \{n_1, n_2, n_3, \dots, n_N\}; N = |V| \quad (1)$$

$$E = \{e_1, e_2, e_3, \dots, e_L\}; e_i \rightarrow (n_i, n_j) \quad \forall n_i, n_j \in V, \quad (2)$$

$$e_i \in E; L = |E|$$

where N and L are the cardinality of the node set and edge set, respectively. Appendix B lists the statistical details of various PTN structures analyzed in the literature. Tables IV to VI provide an empirical comparison of a few network parameters employed in the analysis of PTNs using various spaces of representation, the details of which will be discussed in the subsequent subsections.

A. Connectivity in Public Transport Networks

In a public transport network, the connectivity pattern of a node with its neighbors is evaluated by a network parameter termed *degree*, which is the number of edges incident on a node. Degree is one of the most fundamental, yet significant parameters in network analysis. Degree is a local property of a node, and average degree of a network is a global parameter which conveys information on the average connectivity of nodes in the entire network. Depending on the graph type, the degree, k , and average degree, $\langle k \rangle$, for undirected networks are defined as

$$k_i = \sum_{j=1}^N a_{ij} \quad \forall i, j \in V, i \neq j, \quad \langle k \rangle = \frac{1}{N} \sum_{i=1}^N k_i \quad (3)$$

For directed networks they are

$$k_i^{\text{in}} = \sum_{j=1}^N a_{ji}, \quad k_i^{\text{out}} = \sum_{i=1}^N a_{ij}, \quad k_i^{\text{total}} = k_i^{\text{in}} + k_i^{\text{out}} \quad \forall i, j \in V, i \neq j \quad (4)$$

$$\langle k^{\text{in}} \rangle = \frac{1}{N} \sum_{i=1}^N k_i^{\text{in}}, \quad \langle k^{\text{out}} \rangle = \frac{1}{N} \sum_{i=1}^N k_i^{\text{out}}, \quad \langle k^{\text{total}} \rangle = \langle k^{\text{in}} \rangle + \langle k^{\text{out}} \rangle \quad (5)$$

All symbols in equations (3)–(5) are defined in Table 3.

Table III.
Nomenclature List.

| Notation | Details |
|--------------------------------------|--|
| A | Adjacency matrix |
| a_{ij} | An element of A , defining the directed connectivity between nodes i and j |
| a_1 | Percentage of total population accessing stops on layer α |
| a_2 | Percentage of total population accessing stops on layer β |
| C_i | Local clustering coefficient of node i |
| $\langle C \rangle$ | Average clustering coefficient |
| C_Δ | Global clustering coefficient |
| $C_b(i)$ | Betweenness centrality of a node i |
| $C_b(e_{im})$ | Betweenness centrality of an edge e_{im} connecting nodes i and m |
| $C_c(i)$ | Closeness centrality of a node i |
| c_d | Cost of a shortest path d |
| $\langle d \rangle$ | Average path length between two nodes |
| $\langle d_{tr} \rangle$ | Average path length between two nodes considering the number of transfers |
| d_{ij} | Geodesic path or shortest path between nodes i and j |
| $d_{ij}(k)$ | Shortest path between nodes i and j through the node k |
| d_{\max} | Diameter of the network |
| d_m | # of points-of-interests of category m |
| $DF_{R'}$ | Duplication factor of a bus route R' |
| E or M | Set of edges in a network |
| k_i | Degree of a node i |
| k_{i_α} | Degree of a node i on layer α |
| k_{\max} | Maximum degree of a node i |
| k_{\min} | Minimum degree of a node i |
| k_i^{in} | In-degree of a node i in a directed network |
| k_i^{out} | Out-degree of a node i in a directed network |
| k_i^{total} | Total degree of a node i in a directed network |
| k_i^w | Weighted degree of a node i |
| $(k_i^{\text{in}})^w$ | Weighted in-degree of a node i in a directed network |
| $(k_i^{\text{out}})^w$ | Weighted out-degree of a node i in a directed network |
| $(k_i^{\text{total}})^w$ | Weighted overall degree of a node i in a directed network |
| $\langle k \rangle$ | Average degree of an undirected network |
| $\langle k^w \rangle$ | Average weighted degree of an undirected network |
| $\langle k^{\text{in}} \rangle$ | Average in-degree of a directed network |
| $\langle (k^{\text{in}})^w \rangle$ | Average weighted in-degree of a directed network |
| $\langle k^{\text{out}} \rangle$ | Average out-degree of a directed network |
| $\langle (k^{\text{out}})^w \rangle$ | Average weighted out-degree of a directed network |
| $\langle k^{\text{total}} \rangle$ | Average overall degree of a directed network |

(Continued)

Table III.
Nomenclature List. (Continued)

| Notation | Details |
|---|--|
| $\langle (k^{\text{total}})^w \rangle$ | Average weighted overall degree of a directed network |
| L | Cardinality of edges in a network, i.e., $L = E $ |
| L_{proj} | Link projected graph of a bipartite graph |
| n_i^* | i th node in a network |
| N | Cardinality of nodes in a network i.e. $N = V $ |
| N_k | Number of nodes with degree k |
| N_{proj} | Node projected graph of a bipartite graph |
| p_k | Probability of finding a node with degree k |
| P_i | # of people accessing stop i |
| P_α | # of people accessing the stops on layer α |
| P_β | # of people accessing the stops on layer β |
| P_T | Total population |
| R | The number of bus routes a stop joins |
| R' | Number of routes operating between two nodes |
| S | The number of stops in a bus route |
| tr_{ij} | Number of transfers between nodes i and j |
| V | Set of nodes in a network |
| v_{ij} | Average vehicular speed along an edge connecting nodes n_i and n_j |
| w_{ij} | Weight of an edge connecting nodes i and j |
| $(w_{i\alpha})_Z$ | Weight of a node i on layer α in a zone Z |
| w_i | Overall weight of a node i |
| λ | Poisson coefficient |
| α | Exponential coefficient |
| γ | Power law coefficient |
| γ_{in} | Power law coefficient for in-degree in a directed network |
| γ_{out} | Power law coefficient for out-degree in a directed network |
| r | Assortativity coefficient |
| $r^{(2)}$ | Assortativity coefficient for second neighbors of a node |
| σ | Small-world parameter |
| ω | New small-world parameter |
| ρ_{P_α} | Density of people accessing stops on layer α in a zone Z |
| ρ_{N_α} | Density of stops layer α in a zone Z |
| $\frac{\rho_{P_\alpha}}{\rho_{N_\alpha}}$ | Node occupying probability (NOP) |
| ϵ | Length difference between the two routes |
| λ_{th} | Route length divergence threshold |
| γ' | Transfer count difference |
| ξ | Route transfer count divergence |

*a node n_i is interchangeably represented as i in a few sections for brevity.

| Appendix B: Statistical Details of PTNs | | | | | | | |
|---|---------|------|------|--------|------------------|-------------|------|
| City | Place | Mode | N | Routes | Spatial Analysis | Space Type | Ref. |
| Debrecen | Hungary | BET | 306 | 53 | ✓ | L-space | [12] |
| Gyor | | B | 230 | 43 | | | |
| Miskolc | | BT | 257 | 35 | | | |
| Pécs | | B | 256 | 55 | | | |
| Szeged | | BET | 242 | 40 | | | |
| Ahmedabad | India | B | 1103 | • | • | L, P-space | [22] |
| Chennai | | | 1009 | | | | |
| Delhi | | | 1557 | | | | |
| Hyderabad | | | 1088 | | | | |
| Kolkata | | | 518 | | | | |
| Mumbai | | | 2267 | | | | |
| Beijing | China | B | 7864 | 1308 | ✓ | L, P-space | [10] |
| Shanghai | | | 5931 | 842 | | | |
| Hangzhou | | | 2750 | 509 | | | |
| Shanghai | China | B | 9502 | 1641 | • | P-space | [24] |
| Beijing | | | 9361 | 1714 | | | |
| Guangzhou | | | 3891 | 1256 | | | |
| Shenzhen | | | 3594 | 884 | | | |
| Dongguan | | | 3269 | 346 | | | |
| Chengdu | | | 3053 | 505 | | | |
| Foshan | | | 2952 | 378 | | | |
| Hangzhou | | | 2789 | 688 | | | |
| Tianjin | | | 2721 | 552 | | | |
| Suzhou | | | 2662 | 341 | | | |
| Beijing | China | B | 3938 | 516 | • | L-, P-space | [20] |
| Shanghai | | | 2063 | 501 | | | |
| Nanjing | | | 1150 | 174 | | | |
| Warsaw | • | BTM | 1533 | • | ✓ | L-, P-space | [15] |
| Switzerland | | | 1613 | | | | |
| Europe | | | 4853 | | | | |
| Pila | Poland | BT | 152 | • | • | L, P-space | [16] |
| Belchatów | | | 174 | | | | |
| Jelenia Góra | | | 194 | | | | |
| Opolew | | | 205 | | | | |
| Toruní | | | 243 | | | | |
| Olsztyn | | | 268 | | | | |
| Gorzów Wlkp | | | 269 | | | | |
| Bydgoszcz | | | 276 | | | | |
| Radom | | | 282 | | | | |
| Zielona Góra | | | 312 | | | | |

(Continued)

Appendix B: Statistical Details of PTNs (*Continued*)

| City | Place | Mode | N | Routes | Spatial Analysis | Space Type | Ref. |
|--------------|--------------|-------------|----------|---------------|-------------------------|-------------------|-------------|
| Gdynia | Poland | | 406 | | | | |
| Kielce | | | 414 | | | | |
| Czestochowa | | | 419 | | | | |
| Szczecin | | | 467 | | | | |
| Gdańsk | | | 493 | | | | |
| Wrocław | | | 526 | | | | |
| Poznań | | | 532 | | | | |
| Białystok | | | 559 | | | | |
| Kraków | | | 940 | | | | |
| Lódź | | | 1023 | | | | |
| Warszawa | | | 1530 | | | | |
| GOP | | | 2811 | | | | |
| Hangzhou | China | B | 827 | 150 | • | P-space | [23] |
| Nanjing | | | 1764 | 252 | | | |
| Beijing | | | 4199 | 572 | | | |
| Shanghai | | | 4374 | 968 | | | |
| Baoding | China | B | 634 | 52 | • | P-space | [19] |
| Jinan | | | 883 | 100 | | | |
| Shijiazhuang | | | 1299 | 139 | | | |
| Berlin | • | BMTU | 2992 | 211 | • | L-, P-, C-space | [17] |
| Dallas | | B | 5366 | 117 | | | |
| Düsseldorf | | BMT | 1494 | 124 | | | |
| Hamburg | | BFMTU | 8084 | 708 | | | |
| Hong Kong | | B | 2024 | 321 | | | |
| Istanbul | | BMT | 4043 | 414 | | | |
| London | | BMT | 10937 | 922 | | | |
| Los Angeles | | B | 44629 | 1881 | | | |
| Moscow | | BEMT | 3569 | 679 | | | |
| Paris | | BM | 3728 | 251 | | | |
| Rome | | BT | 3961 | 681 | | | |
| Sao Paolo | | B | 7215 | 199717 | | | |
| Sydney | | B | 1978 | 596 | | | |
| Taipei | | B | 5311 | 389 | | | |
| Curitiba | Brazil | BM | 9423 | 615 | ✓ | P-space | [25] |
| Qingdao | China | B | 1758 | 261 | ✓ | L, P-space | [19] |
| Beijing | China | B | 5421 | 722 | • | L-, P-space | [18] |
| Harbin | China | B | 993 | 132 | • | P-space | [26] |
| Singapore | • | B | 4620 | 428 | ✓ | C-space | [27] |
| Nagoya | Japan | BM | 687 | 280 | • | L-, P-, C-space | [37] |

B: Bus, E: Electric trolleybus, T: Tram, M: Metro (subway), U: Urban train, F: Ferry.

The weighted node degree and the average weighted node degree are defined similar to (3)–(5), where a_{ij} is multiplied by w_{ij} , the edge weight (to be discussed in Section IV-J). Furthermore, Tables IV to VI tabulate the empirical values of average node degree under various spaces of representation. From Table IV, we observe that the average node degree in L-space analysis is nearly equal to two (in general) indicating that a stop is merely connected to its neighboring stops. On the other hand, the values shown in Table V indicate that the average

node degree in the P-space analysis is roughly 10 times higher than that in the L-space which denotes the average number of nodes that can be reached from a certain node with or without making a transfer. Appendix C lists the various interpretations of the node degree under different spaces of network representation. The key point is that significant features like connectivity of a node in the L-space representation, route overlapping pattern in the C-space representation, and the number of transfers to be made in the P-space representation can be more

Table IV.
Empirical values of various network parameters in L-space representation.

| $\langle k \rangle$ | C_Δ | $\langle d \rangle$ | r | References |
|--------------------------------|---------------|---------------------|----------|------------|
| Bus Transport Network | | | | |
| 2.48–3.03 | 0.055–0.161 | 6.83–21.52 | +ve | [16] |
| 2.88–4.59 | 0.09–0.15 | 7.13–12.56 | +ve | [20] |
| 2.1–2.4 | 0.0004–0.0129 | 28.1–50.9 | • | [15] |
| 1.18–3.59 | • | 6.4–52 | +ve | [17] |
| 3.13 | 0.142 | 20.03 | +ve | [18] |
| 2.25–2.50 | 0.06–0.08 | 21.09–43.02 | • | [10] |
| • | • | 10.8–14.5 | • | [12] |
| 3.67–24.58 | 0.07–0.26 | 3.87–25.69 | +ve, -ve | [11] |
| 2.65–2.92 | • | • | +ve | [19] |
| 2.65–2.92 | 0.05–0.09 | 13.82–20.9 | • | [21] |
| 1.91–3.77 | 0.074–0.213 | 9.9–102 | • | [13] |
| Metro Transport Network | | | | |
| • | • | 10.74–15.60 | • | [32] |
| 2–2.45 | 0–0.077 | 10–16 | • | [29] |
| 2.2 | 0.0018 | • | • | [31] |
| • | 0.390–0.710 | • | • | [34] |
| 2–2.4 | • | 6.7–19.9 | • | [36] |
| • | • | 10.13–15.02 | • | [32] |

Table V.
Empirical values of various network parameters in P-space representation.

| $\langle k \rangle$ | C_Δ | $\langle d \rangle$ | r | References |
|------------------------------|---------------|---------------------|----------|------------|
| Bus Transport Network | | | | |
| 33.13–90.93 | 0.682–0.847 | 1.71–2.90 | +ve, -ve | [16] |
| 41.06–94.19 | 0.73–0.78 | 2.54–2.66 | +ve, -ve | [20] |
| 24.6–102.3 | 0.6829–0.9095 | 2.3–3.7 | • | [15] |
| 4–11 | • | 2.2–4.7 | +ve, -ve | [17] |
| 44.60–122.89 | 0.716–0.819 | 2.84–3.45 | +ve, -ve | [24] |
| 35.84–60.24 | 0.57–0.68 | 3.15–3.46 | • | [10] |
| 44.40–92.54 | 0.69–0.81 | 2.42–3.45 | • | [25] |
| 44.46–134.65 | 0.73–0.78 | 2.53–2.89 | • | [23] |

readily identified via studying the node degree. In addition, the study of the degree distribution in a network would benefit the evaluation of an interesting network property called the *scale-free* property.

B. Are Public Transport Networks Scale-Free?

Following the random network model proposed by Paul Erdős and Alfréd Rényi [38], many real-world networks were verified to be connected in a random way, in which

a myriad number of nodes in the network exhibit similar degree since the nodes are connected randomly. The degree distribution of such a random network is more likely to follow a Poisson distribution [38], [39]. However, Barabási [2], [5], [40], [41] showed a unique behavior in which a few nodes in the network exhibit very high degree while a large number of nodes exhibit low degree, and the degree distribution of such network is expected to

Table VI.
Empirical values of various network parameters in C-space representation.

| $\langle k \rangle$ | C_Δ | $\langle d \rangle$ | r | References |
|------------------------------|------------|---------------------|-----|------------|
| Bus Transport Network | | | | |
| 11.09–151.72 | 2.14–28.3 | 1.7–4 | +ve | [17] |
| 98.1 | • | • | • | [27] |

Appendix C: Network Parameters in Different Spaces of Representation

| Parameter | L-space | P-space | C-space | B-space |
|---------------------------------|--|--|---|--|
| Degree | Number of neighboring stops that a given stop is connected to | Number of stops accessible from a given stop with or without making a transfer | Number of overlapped routes | Number of stations serviced by a route (in L_{proj} graph) or number of routes a station is connected to (in N_{proj} graph) |
| Local clustering (transitivity) | Cohesiveness among the neighbors of a node considering the physical infrastructure | Cohesiveness among the neighbors of a node considering the actual connectivity | Cohesiveness among the neighbors of a node considering the common stops serviced along the routes | Cohesiveness between the routes and stops in a network |
| Average path length | Total number of links (hops) to be traversed between the chosen O-D | Total number of transfers to be taken to travel between the chosen O-D | – | – |
| Betweenness centrality | Node significance based on the number of shortest path routes that can traverse via the given node | Node significance based on the number of transfers than can be handled by the given node | – | – |
| Closeness centrality | Reachability of a node with respect to every other node in the network | Reachability of a node with respect to other routes in the network considering the number of transfers | – | – |
| Assortativity | Correlation level between similar degree stops in the network | Correlation level between similar degree routes in the network | Correlation between similar degree routes based on their overlapping | – |
| Communities | Identifying different zones in the network based on a behavior of the stops and their connectivity | Identifying different zones in a network based on the behavior of the routes | Identifying different zones in the network based on the behavior of route overlapping | – |

follow a power law distribution. Such networks are called scale-free networks. Observing the scale-free property in public transport networks can be inspiring since it demonstrates a strong prevalence of the hierarchical network structure, i.e., hubs at the top of the hierarchy serve maximum demand, while those below are relatively modest nodes serving mediocre demand. Intuitively, although we would expect a certain number of stops in a network are serviced by a large number of routes, it is intriguing to verify such property mathematically. Interestingly, it was observed that some of the public transport networks do exhibit the scale-free property. Furthermore, as explained later in this section, the degree distribution in a network is a good source of inference on the network evolution [23], [42]. Thus, the study of degree distribution has attracted enormous research interest.

The degree distribution exposes the probability of a randomly selected node in the network having a degree of k , i.e.,

$$p_k = \frac{N_k}{N} \quad \text{or} \quad N_k = Np_k \quad (6)$$

where p_k is the probability of finding a node with degree k , N_k is the number of nodes with degree k , and N is the total number of nodes in the network. Interested readers may refer to [5, Chapters 3–5] to probe further into the difference between random and scale-free networks. Table VII tabulates the degree distributions of various PTNs reported in the literature. From Table VII, we make the following observations:

- i) An exponential degree distribution in L-space indicates that connecting a newly added node with

the existing nodes is more likely to be random. This is in contrary to the notion of preferential attachment where newly added nodes are connected to the already existing influential nodes in the network, making the degree distribution a power-law distribution.

- ii) An exponential degree distribution in P-space indicates that defining a new route sequence in the network is more likely to be random in order to ensure a better coverage and service rather than along the influential nodes in the network.
- iii) An exponential degree distribution in C-space indicates that defining the stops along a route node is more random than defining the stops along a route to cover the influential nodes.

Thus, the degree distribution of a network provides information on the topological evolution of the public transport network in a city [23]. Up to now, some simple network evolution models have been proposed based on fitting empirical data. However, the nature of network evolution has never been verified from the actual deployment perspective. As demonstrated by Barabási [2], the existence of hubs in a scale-free network can be a result of two phenomena, namely, growth and preferential attachment. However, the feasibility of deployment of preferential attachment in a real-world network is yet to be verified!

In our previous work [13], as part of analyzing bus transport networks, we proposed a supernode graph representation, where a supernode is a cluster of geographically closely-located nodes which satisfy the criterion $d_{th} \leq 100$ m, where d_{th} is the geographical distance between two nodes. Using the supernode representation, we analyzed

Table VII.
Degree distribution patterns from some public transport network analyses.

| L-space | P-space | C-space | References |
|--------------------------------|-------------------|-------------|----------------|
| Bus Transport Network | | | |
| Power law | Exponential | • | [16] [20] |
| Shifted power law | • | • | [18] |
| Power law | • | • | [13] |
| Power law | Shifted power law | • | [19] |
| Heavy tailed | Power law | • | [11] [37] |
| Exponential | Exponential | • | [10] [15] [17] |
| Exponential | • | • | [12] [26] |
| • | Exponential | • | [23] [24] |
| • | Power law | • | [25] |
| • | • | Exponential | [27] |
| Metro Transport Network | | | |
| Power law | • | • | [33] [34] |

the scale-free behavior for three cities, and it was very interesting to observe that the Hong Kong network plausibly exhibited the scale-free property with the supernode representation, as shown in Fig. 2. In other words, a slight modification in the topological representation permitted the exposition of an important network property which otherwise was undetected under conventional graph representation. Therefore, the effect of supernodes in analyzing the public transport networks should not be overlooked.

Finally, it is very interesting to observe the scale-free property (sometimes called the 80/20 rule) in public transport networks. This demonstrates the fact a myriad number of stops carry 20% of the network load, and a countable number of stops carry 80% of the load. Public transport networks having such a property are free of any scaling applied to them. The mechanism of passenger flow in a scale-free network is an important research topic from the perspective of a transport engineer, similar to the study of information spread or disease spreading by network engineers and biologists. Another core research area of practical importance is robustness analysis which aims to study the network functionality upon removal of a certain set of target

nodes. It has been shown that scale-free networks are more prone to targeted attacks, in contrary to random networks which end up merely at network fragmentation on targeted attacks.

C. Network Cohesiveness

The extent to which the immediate neighbors of a node are connected to each other is examined through a property called *clustering*, which defines the level of cohesiveness in a network. Clustering, also known as the transitivity, is a local property dealing with node level information in network theory. The cohesiveness of nodes is evaluated at local level through a parameter called *local clustering coefficient*, which is given by

$$C_i = \frac{\sum_{j,h} a_{ij} a_{ih} a_{jh}}{k_i(k_i - 1)} \quad (7)$$

for undirected networks, and

$$C_i = \frac{\sum_j \sum_h (a_{ij} + a_{ji})(a_{jh} + a_{hj})(a_{hi} + a_{ih})}{2[k_i(k_i - 1) - 2k_i^-]}; \\ k_i^- = \sum_{i \neq j} a_{ij} a_{ji} \quad (8)$$

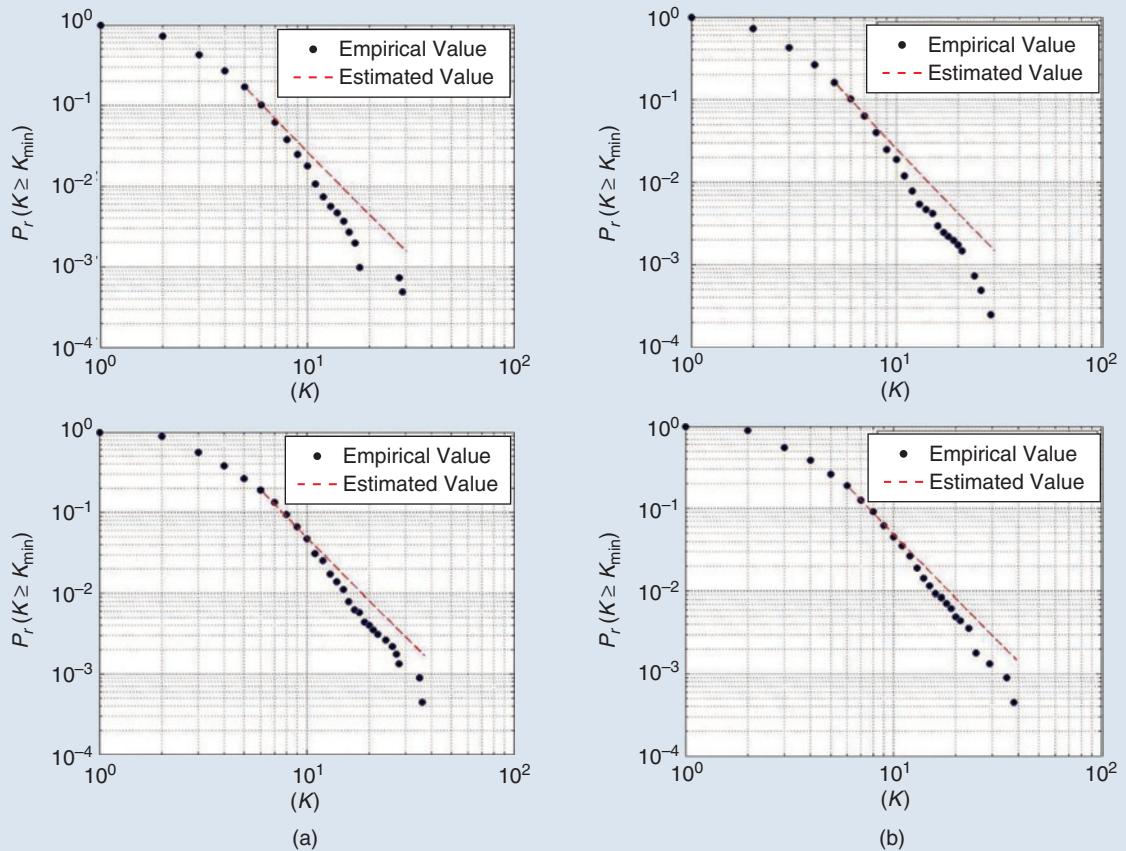


Figure 2. Power law fit for (a) in-degree distribution; (b) out-degree distribution under regular (upper) and supernode (lower) representations.

for directed networks. At the global level, the *global clustering coefficient* is given by

$$C_{\Delta} = \frac{1}{N} \sum_i C_i. \quad (9)$$

Again, all symbols are defined in Table 3. For an in-depth discussion of evaluating clustering by identifying triads or cliques in different graph types, interested readers are referred to ref. [43].

The study of clustering coefficients by itself has not attracted much attention from researchers in the analysis of PTNs. However, some inspiring observations can be made from the relationship between C_i and k .

- i) The dependency of C_i and k closely resembles a power law where the value of C_i for a given $k(C_i(k))$ is close to unity for small values of k , and $C_i(k)$ decreases rapidly with increasing k [11], [16], [20].
- ii) As observed from (7) and (8), the inverse dependency of C_i on k indicates the hierarchical structure of a network in the L-space representation, where high degree nodes (hubs) tend to form numerous connections with their neighbors, thus reducing the possibility of their neighbors having connections among themselves. This reduces the local clustering coefficient of high degree nodes. On the other hand, a low degree node has a greater tendency to be connected among its neighbors, increasing its local clustering coefficient [11].
- iii) In the P-space representation, all stations of a specific route form a perfect clique, with $C_i = 1$ for all nodes in the route. The value of C_i becomes smaller when the nodes are shared by multiple routes. Thus, in the P-space representation, the fully connected subgraphs of all stops along a route constitute local cliques, and these local cliques are shared between routes through the common nodes. Hence, the nodes with a low degree and a high clustering coefficient belong to a fully connected local clique, whereas the nodes with a high degree and a low clustering coefficient connect multiple local cliques, reflecting that hubs act as coordinating points for several routes [16], [23]–[25], [44]. Thus, the distribution of $C_i(k)$ gives an indication on how the clustering is organized for nodes of various degrees.

Appendix C summarizes the common interpretations of transitivity under various spaces of network representation, and Tables IV to VI give the ranges of values of the global clustering coefficient under the various spaces of network representation. It can be seen that the clustering in P-space is significantly higher than that in L-space due to the existence of more local cliques in P-space. Although clustering has been extensively employed in

L-space PTN analysis, the physical significance of evaluating both local and global clustering coefficients in L-space is vague. Moreover, the clustering coefficient is more meaningfully interpreted in the P-space representation for a PTN analysis. Also, evaluating the clustering coefficient in B-space (bipartite graph) is meaningless since the neighbors of a node are from the same group, and there exists no connection between nodes of the same group in B-space. However, evaluating clustering in C-space conveys interesting information on the extent of route overlapping in a network which is an extremely useful information for route optimization, and thus deserves more work.

D. Travel Distance in Hops

In a PTN, the number of hops to be traversed to accomplish a journey between any two chosen stops in a network is normally measured by *path length*. In graph theory, a path is a sequence of nodes connected by links. The *shortest path length* is the shortest number of links between two chosen nodes, and the *average path length* (geodesic path) is the average of the shortest path length between all node pairs in the network. The *diameter* is the longest of all shortest paths, and is an upper bound of the average path length. Although the measure of path length conveys no information on the number of transfers to be made during the journey, it is still an important measure in the public transport network analysis from a passenger point of view since the number of hops is definitely one of the prime factors considered by the passengers in selecting a route for the journey. There are a few notable algorithms for finding the average path length in a network [45]. However, it should be noted that the edge weight should be cautiously chosen (represented) in the evaluation of the average path length in a weighted graph in order to avoid a wrong interpretation of the measured path length. For example, the Dijkstra's algorithm using d_{ij} (geographical distance between two stops) and v_{ij} (average vehicular speed along a road segment) as the edge weight may generate two completely different results in evaluating the path length between two chosen nodes [46]. The average shortest path length is usually given by

$$\langle d \rangle = \frac{\sum_{i \neq j} d_{ij}}{N(N-1)} \quad \forall i = j = 1, 2, \dots, N \quad (10)$$

where d_{ij} is the geodesic distance between nodes n_i and n_j . Also, $d_{ij} = 1$ if there exists a path between the two nodes, and $d_{ij} = \infty$ otherwise, implying a possible divergence problem in a non-connected graph. A smaller value of d indicates a shorter travel distance (with or without transfers) that a passenger should take to

accomplish a journey. The different perspectives of average path length are given in Appendix C. A detailed comparison of average path length in different spaces has been given in Tables IV to VI. From the values of $\langle d \rangle$ given in Tables IV to VI, it is evident that the average

path length in the L-space representation is significantly longer than that in the P-space representation. Thus, the average number of links traversed by a user is much larger than the number of transfers made to reach the destination. A few other notable observations concerning the average path length are

- i) An inhomogeneous distribution of stops within a city leads to Gaussian or asymmetric unimodal distribution (with longer tail ends) in the L-space and P-space representations. Thus, a fewer number of stops in the suburbs/downtown in a city leads to long travel distances. This accounts for the long tail ends in the distribution. This phenomenon is consistent with the plethora of stops observed at city centers leading to short travel distances [12], [16], [17]. A rather unique feature can be observed in the distribution pattern in ref. [17], where a secondary peak in the tail end of the distribution along with the major peak has been observed, indicating that in addition to a major central business district (CBD), a supporting minor CBD exists in the city.
- ii) As studied in ref. [15], the average path length of a network is significantly affected in L-space by the existence of shortcut paths. Despite the absence of physical connectivity between a few nodes in the PTN (e.g., between a bus stop and a metro station which are geographically close, or stops on either sides of a road segment), they can be virtually connected by a short walking distance and such nodes can be represented as short distance station pairs (SSPs) or supernodes. Thus, merely representing the physical connectivity of two different transportation networks does not justify the true measure of the average path length [10–13]. However, a slight reorganization of the network topology using supernodes provides a better and more practical insight on the average path length estimation in PTN analysis [10], [13].
- iii) Fig. 3 shows the path length distribution of bus stops for the three cities analyzed in our previous work with and without considering supernodes in the network [13]. For all the three cities in Fig. 3, it has been observed that the path length values are comparatively small when the supernode representation is used which conveys more clear information on the actual path length to be traversed in practice. Thus, in a PTN analysis, the supernode representation offers a more realistic path length estimation.
- iv) The link length distribution (the distribution of geographical distance between the stops) conveys captivating information on the route length adopted by public transport networks. In ref. [10],

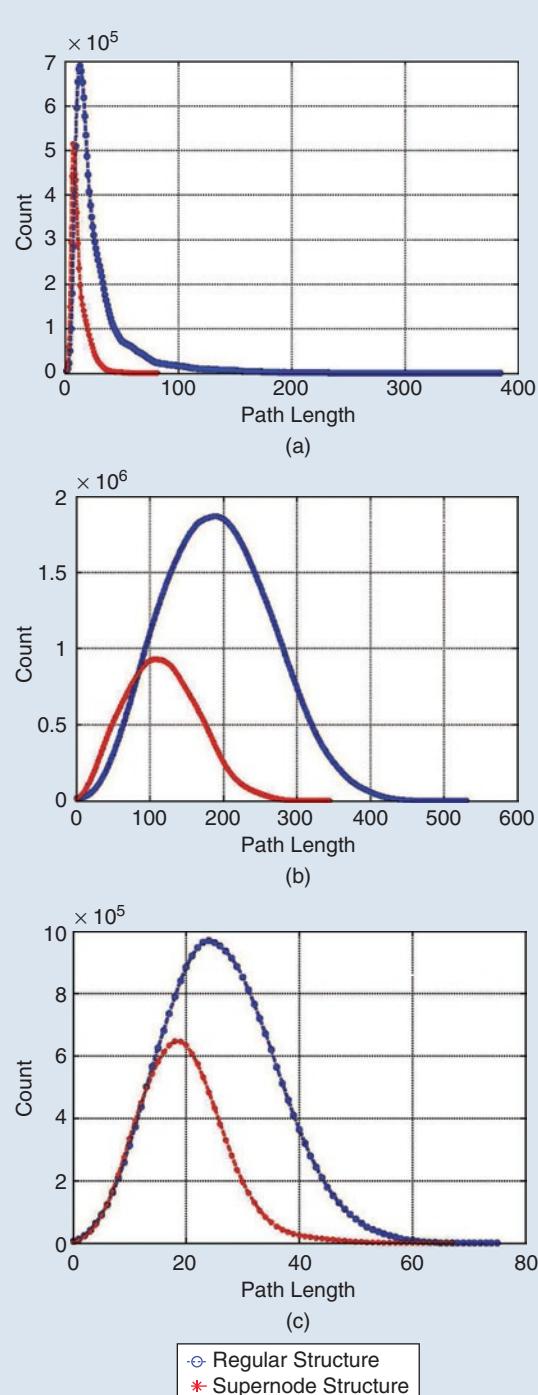


Figure 3. Average path length distribution for (a) Hong Kong; (b) London; and (c) Bengaluru networks with and without considering supernodes.

the geographic link length distribution has been found to follow a power law, indicating that a substantial number of routes in the public transportation have a short geographical route length and only a nominal number of routes have a long route length. Furthermore, such an analysis sheds useful light on the city's demographics. (Note: Since the latitude and longitude information of the stops are given in a spherical coordinate system, the great-circle distance is preferred over the Euclidean distance in evaluating the geographic distance between two stops [47]).

- v) In PTN analysis, the average shortest path length between any two nodes in the network might not always guarantee a minimum number of transfers. Hence, combining the number of transfers with the shortest path length offers a more realistic choice for traveling between a chosen node pair. Zhang [18] has demonstrated a way of measuring the shortest path length in (10) taking into consideration the number of transfers along the shortest path, i.e.,

$$\langle d_{tr} \rangle = \frac{\sum_i \sum_j d_{ij}(1 + tr_{ij})}{N(N-1)} \quad \forall i = j = 1, 2, \dots, N \quad (11)$$

where tr_{ij} is the total number of transfers needed to travel between nodes i and j .

E. Small-worldness in Public Transport Networks

First demonstrated by Watts and Strogatz [48], a class of networks, called *small-world* networks, exhibit high clustering and a low average path length. Empirically the small-world property of a network can be verified by

$$\sigma = \frac{C}{\frac{d}{d_{rand}}} = \frac{\gamma}{\lambda} \quad (12)$$

where C_{rand} and d_{rand} are the clustering coefficient and average path length values of the equivalent random networks (degree conserved network of the same size) [49]. If $\sigma > 1$, i.e., when $C \geq C_{rand}$ and $d \approx d_{rand}$, the network can be classified as a small-world network. Telesford et al. [50] pointed out that the comparison of average path length of a given network to its equivalent random network is acceptable; however, the comparison of clustering of a network to that of its equivalent random network does not fully capture the small-world behavior since the clustering of a network is expected to behave close to a lattice structure. It is also observed in (12) that even a small change in C_{rand} will affect the value of the small-world parameter (σ). Hence, a new approach to capture the small-worldness of a network can be adopted, as proposed by Telesford et al. [50], i.e.,

$$\omega = \frac{d_{rand}}{d} - \frac{C}{C_{latt}} \quad (13)$$

where C_{latt} and d_{rand} are the clustering coefficient and average path length values of the equivalent lattice and random network, respectively. In (13), when $C \approx C_{latt}$ and $d \approx d_{rand}$, we have $\omega \approx 0$ and such networks are considered small-world networks. By simulating the behavior of a small network, Telesford et al. [50] demonstrated the variation of σ and ω , where $\sigma > 1$ for all values of p (except $p = 1$). This means that the network would show the small-world property for all the rewiring probabilities (except $p = 1$), demonstrating that $\sigma > 1$ cannot fully capture the small-worldness. However, the variation of ω shows three major zones, viz. $\omega < 0$, $\omega \approx 0$, and $\omega > 0$, capturing the random, small-world, and lattice properties of the network [50]. Furthermore, interested readers may refer to refs. [48], [49] for details on the basic rewiring approaches.

Some reported works have attempted to use (12) to test the small-worldness of public transport networks by verifying $\sigma > 1$, but such results have been found to deliver misleading conclusions [11], [16], [24], [25], [37]. In our previous work [13], we adopted Telesford et al.'s method to evaluate the small-world property of bus transport networks, and the results of two networks are shown here in Fig. 4. By observing the value of ω in Fig. 4(a) we can see that the Hong Kong network becomes a small-world network if certain modifications are made to the existing routes. However, from the value of ω shown in Fig. 4(b), we can also see that the modifications in the routes needed can be quite substantial and hence difficult to implement.

Unlike Stanley Milgram's experiment conducted in 1967 for studying the small-world behavior of a social network [51], finding a small value of average path length in large public transport networks is much more difficult. In addition, it is widely known that $\langle d \rangle$ varies with \sqrt{N} [5]. Thus, a true measure of small-worldness should consider the network size as one of the parameters alongside with the clustering and average path length. Small-worldness is undoubtedly an important network behavior in public transport networks as it demonstrates the effectiveness of a transport network in terms of both connectivity (clustering) and the travel distance in hops (path length). However, existing measures of small-worldness have merely been used to demonstrate high clustering and low average path length, and a practical measure from the passenger's perspective would be more desirable for public transport networks.

F. Bridges in Public Transport Networks

Centrality is a network parameter describing primarily local information about nodes (edges), and yet having a

global significance. Centrality quantifies the significance of a node (edge) based on various sources of information. Centrality measures may thus include degree centrality, Eigen-centrality, Katz-centrality, page rank centrality, closeness centrality, betweenness centrality, etc. In PTN analysis, a few centrality measures have been extensively studied, e.g., degree centrality, closeness centrality, and betweenness centrality. The degree

centrality, as discussed in Section IV-A, rates a node's significance according to its degree. Similarly, betweenness centrality emphasizes the capability of a node in bridging multiple shortest paths in a network [52]. Specifically, the *node betweenness centrality* is defined as

$$C_b(i) = \sum_{i,j,k \in V} \frac{d_{jk}(i)}{d_{jk}}, \quad (14)$$

and the *edge betweenness centrality* is defined as

$$C_b(e_{im}) = \sum_{i,j,k,m \in V} \frac{d_{jk}(e_{im})}{d_{jk}} \quad (15)$$

where d_{jk} is the total number of shortest paths between nodes j and k , and $d_{jk}(i)$ or $d_{jk}(e_{im})$ is the shortest paths between nodes j and k passing through node i or edge e_{im} . Appendix C summarizes the different perspectives of betweenness centrality under various spaces of network representation.

For a given network, it is intuitive to assume that the nodes having a higher degree have a higher probability to serve as central nodes in the network, and thus, the relationship between degree and betweenness centrality has been actively studied. The major observations are as follows:

- The dependency of betweenness upon degree is found to follow a Poisson distribution in the L-space representation [16], and a power-law distribution in the L-space representation [22] and the C-space representations [17].
- In the P-space representation, two variations of power law distribution have been observed depending on the value of k . For small values of k , the betweenness is almost zero leading to a steep slope in the power-law distribution, whereas for high values of k , a larger betweenness has been observed, leading to a more regular power-law distribution pattern [16], [17].
- In the B-space representation, the distribution pattern is found to be similar to that of the P-space representation since, N_{proj} nodes have low degree and L_{proj} nodes have high degree [17]. Furthermore, Bona *et al.* demonstrated that, the nodes having a high betweenness centrality are mostly situated in CBDs [25]. However, this observation remains partially true because a node in the downtown/suburb which acts as an entry or exit point for passengers traveling between the cities might also contribute to a high betweenness centrality.

In an earlier work [53], we employed betweenness centrality as a prime parameter for studying network behavior when the interaction between multiple transport networks (bus and metro, for example) are ignored. Specifically, to demonstrate the unbalanced use or biasness

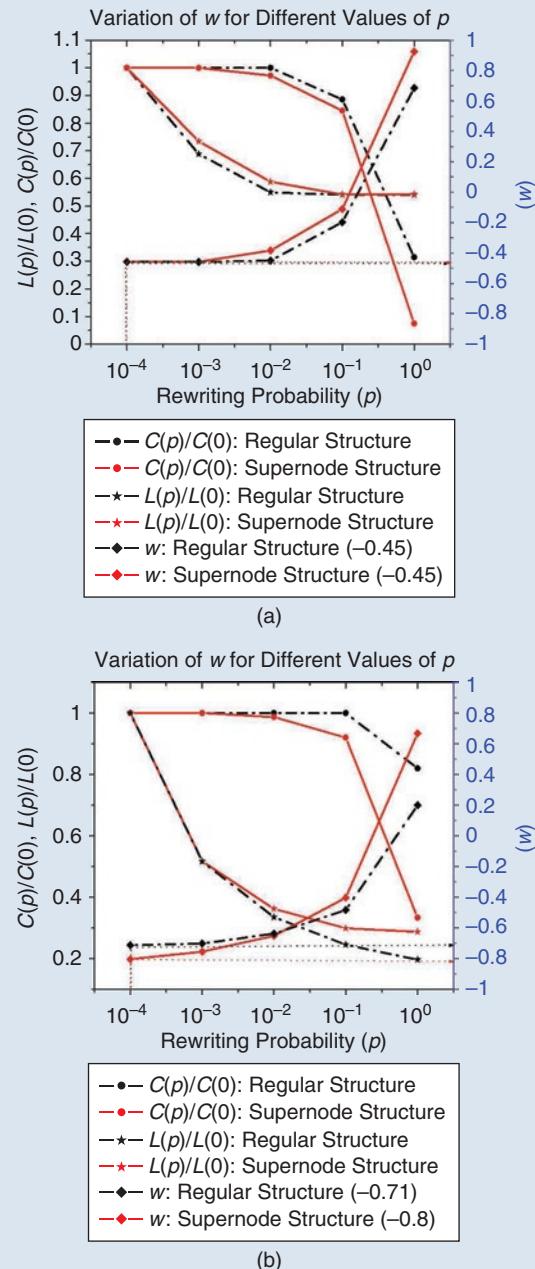


Figure 4. Small world network behavior for (a) Hong Kong; and (b) London networks with and without supernodes, and the value of ω at $p = 10^{-4}$ is highlighted. (Note: L is used instead of $\langle d \rangle$ to represent the average path length).

of PTNs, a node weight was assigned considering the bus (layer α) and metro (layer β) transport layers as individual mono-layers where the layer interaction is ignored. Later, a method of *spatial amalgamation* was applied to integrate the two layers, and accordingly, a new node weight was assigned to the nodes in the integrated multi-layer, i.e.,

$$(P_\alpha = a_1 * P_T)_Z \text{ and } (P_\beta = a_2 * P_T)_Z \quad (16)$$

$$(w_{i\alpha})_Z = \left(\frac{\rho_{P_\alpha}}{\rho_{N_\alpha}} \right)_Z + k_{i\alpha} \quad (17)$$

$$w_i = w_{i\alpha} + C_b(i) \quad (18)$$

where all symbols are listed in Table 3. Similar to equations 16–18, the node weight is evaluated on layer β . Fig. 5 shows the influential nodes ($w_i \geq 0.8$) in the network according to the node weight assigned with and without considering the interaction between the layers. We can see that the assigned node weights differ significantly between the individual mono-layer analysis and the integrated multi-layer analysis. This indicates that ignoring the inter-connectedness between the transport layers leads to a unrealistic conclusions. Betweenness centrality has been employed as the prime parameter for assigning node weight for the multi-layer analysis since passengers may prefer using multiple transport networks (bus and metro) to complete their trips.

One of the main advantages of using betweenness centrality as a measure of significance of a node is that the removal of high betweenness nodes can adversely affect the average path length of the entire network as these nodes essentially control the traffic movement in the network by bridging various routes and nodes. Consideration of betweenness of nodes has recently been incorporated under robustness analysis and is attracting a significant research attention [28], [34], [54]–[57].

G. How Close are the Stops in a Public Transport Network?

Closeness centrality is yet another parameter giving node level information, and in particular indicates how close a node i is to the rest of the network. Normally, closeness is evaluated in terms of hop count, i.e., total number of hops required to reach all other nodes in a network from a given node, i.e., we have

$$C_c(i) = \frac{1}{\sum_j d_{ij}} \quad (19)$$

The smaller the value of d_{ij} , the closer node i is to all other nodes. Prior works [12], [22] have considered the closeness centrality values for weighted and unweighted network structures, respectively, and the corresponding distributions have been found to follow an exponential distribution. Appendix C summarizes the

key perspectives on closeness centrality under various spaces of representation. Due to the limited available results on closeness centrality related to PTNs and the rather restricted analysis in the L-space representation, the practical significance of evaluating closeness centrality of PTNs is still not widely recognized. In addition, in a PTN under the L-space representation, a particular stop is seldom expected to be close to all other remaining nodes in the network as it is typically connected to a portion of the network. However, closeness centrality in other spaces might offer insightful information, and should therefore deserve further investigation.

H. Social Behavior in Public Transport Networks

Observing the social behavior at public transport stops and routes in a PTN is interesting. Specifically, the polarization of connectivity of the stops and routes towards other stops and routes is practically useful. Such social behavior can be studied in terms of *assortativity*. While degree, as discussed in Section VI-A, captures the connectivity of a node in the network, assortativity captures the connectivity among similar kind of nodes in the network. In other words, assortativity reflects the bias of nodes to connect with nodes of similar kind. Thus, assortativity is also a local parameter providing node level

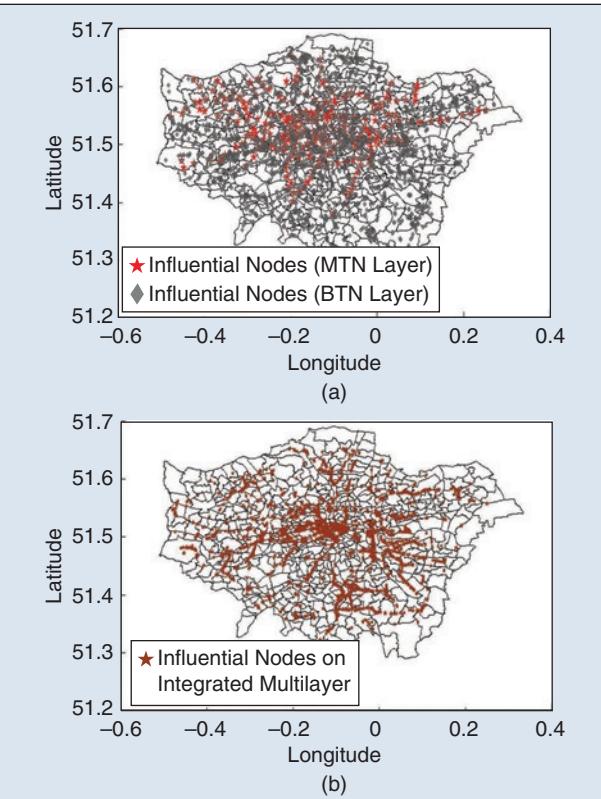


Figure 5. Influential nodes in the London PTN with (a) mono-layer analysis; (b) multi-layer analysis.

information and specifically correlation between node degrees in the network. Depending on the correlation type, the network can be either assortative (connection between two high-degree or low-degree nodes) or disassortative (connection between a high-degree node and a low-degree node). Assortativity can be assessed in terms of the average degree of a node's neighbors [58]. Moreover, Newman [59] later demonstrated that assortativity can be effectively evaluated by the Pearson correlation coefficient, i.e.,

$$r = \frac{M^{-1} \sum_i j_i k_i - \left[M^{-1} \sum_i \frac{k_i + j_i}{2} \right]^2}{M^{-1} \sum_i \frac{j_i^2 + k_i^2}{2} - \left[M^{-1} \sum_i \frac{k_i + j_i}{2} \right]^2} \quad (20)$$

where j_i and k_i are the degrees at both ends of an edge i , M is the number of edges, and $-1 \leq r \leq 1$. The network is assortative if r is +ve, and disassortative if r is -ve. Foster *et al.* [60] extended (20) for a directed network where four typical assortative mixing levels are observed, namely, $r(\text{in, in})$, $r(\text{in, out})$, $r(\text{out, in})$ and $r(\text{out, out})$ denoting the correlation between in-degree of two nodes, out-degree of two-nodes, in-degree of a node, and an out-degree of a node, respectively. The physical significance of assortativity is that a negative value of r shows the existence of core-periphery network structure and a positive value of r shows a layered network structure. In PTN analysis, it is more desirable for the network to be disassortative in order to offer better service and connectivity in a core-periphery structure. However, if a PTN follows a layered architecture, it is desirable to have assortative mixing between highly central nodes or hubs, which in turn are expected to have a disassortative mixing with other nodes in the network.

It has been observed that smaller networks are expected to be more disassortative, and larger networks exhibit both assortative and disassortative tendency [18], [21], [24]. Chatterjee *et al.* [22] developed the degree-correlation matrix to visualize the connectivity preferences of nodes in the L-space and P-space representations. Strong assortativity has been observed in L-space among low degree nodes, whereas, in P-space, strong assortativity can be seen in nodes of certain node degrees. Also, Ferber *et al.* [17] investigated the assortativity for the second neighbor ($r^{(2)}$) of a node, and found that a more positive $r^{(2)}$ indicates stronger correlation with the immediate neighbors as well as the second neighbors. Although the property of assortative mixing has so far been studied with respect to a node degree, the polarization of nodes with respect to other parameters (e.g., various centrality measures) may offer a different perspective in understanding the network behavior. Such study of social behavior of public transport

stops and routes will provide important information for the design of stop locations and route distribution.

I. Communities

Community is a pair-wise parameter studied at node level and yet offers a global view in network theory. Identifying communities in a network, also called network partitioning, can be thought of as an extension to identifying assortative mixing in the network, but over a much larger set of nodes. A community is a subgraph of a network with nodes of similar behavior (in terms of connectivity), and there are dense links within a community but much fewer links between communities. Graph partitioning has been a hot research topic in the field of graph theory in the past decade since evaluating communities, especially in large and dense networks involve computationally intensive processes. An index called modularity is employed to evaluate communities in a network, as demonstrated by Newman and Girvan [61], [62], i.e.,

$$Q = \sum_i s_{ij} - \sum_{ijk} s_{ij}s_{ki} \quad (21)$$

where s_{ij} is a component of matrix s which defines the number of edges in the original network that connects nodes in community i to nodes in community j , and $0 \leq Q \leq 1$. Here, $Q = 0$ indicates the absence of similar degree connectivity in a network (random graph), and $Q = 1$ indicates a strong connection within the communities. Equation (21) has been popularly used to evaluate the modularity index for all types of networks (directed, undirected, weighted and unweighted). Moreover, in the survey conducted by Khan and Niazi [63], various modularity metrics have been considered depending on the network type. In the study by Háznagy *et al.* [12], the city's center has been found to have a few communities whereas the periphery has numerous communities. The work by Bona *et al.* [25] has identified 187 different communities with a modularity value between 0.3 and 0.7 for a PTN in a Brazilian city. For the Chinese city of Qingdao, Zhang *et al.* [19] observed a high modularity value of 0.8 with an average of 20 communities. Furthermore, a total of 46 communities with a strong modularity value of 0.91 was observed in an urban rail transit system in China [18]. Sun *et al.* [27] also found a weak modularity value of 0.34 with 7 communities in urban bus networks, where communities have been consistently identified with respect to their spatial coverage. Appendix C offers various perspectives of understanding community structures under various spaces of network representation. A physical significance of identifying communities in a network is that knowing the structural equivalence of nodes and their communities is crucial to understanding of the behavior of the intra-community and inter-community nodes.

J. Node and Edge Weights

In generating weighted networks, a weight (w) is either added to a node, an edge, or both. Weighted transport networks are still relatively less explored, despite their obvious practical significance in quantifying the relative importance of nodes and edges in relation to the level of service and performance provided by a public transport network. In this section we discuss a few weight metrics commonly employed in the topological analysis of various public transport networks.

Node weight can be assigned to reflect the relative importance of a node (station). For instance, a weight can be assigned to a station or a link according to the number of routes servicing it (degree) [12], [26], or according to the sum of weights of the adjacent edge weights (weighted degree) [27]. Edge weight may be assigned according to the morning peak hour capacity of the vehicles carrying the traffic [12], the minimum geographical distance between any two nodes [10], [21], the number of overlapped bus routes between two stations [11], [21], [27], or the number of common stops serviced along a route in C-space [20]. Furthermore, dynamic edge weights may also be assigned according to the average travel time between two nodes [19], which have been found to be very useful in analyzing the dynamic behavior of PTNs, especially in describing the varying behavior during peak- and off-peak hours.

In our recent work [13], we proposed a static demand estimation approach to assign node weight which reflects the demand centrality of a node, i.e., the capability of a node in serving the static demand by considering the number of *points of interest* (POIs), and the number of people accessing a specific station (node occupying probability). A POI can be a hospital, hotel, office, school, sports arena, cinema, shopping complex or the residential apartment. The crux of this demand estimation approach is that the real-world usage of a bus stop should be strongly dependent on the presence of POIs around the bus stop and the number of people accessing it. Using the information on POIs and node occupying probability (NOP), the node weight is evaluated as

$$w_i = c_1 \left(\sum_{m=1}^4 d_m \right) + c_2 P_i + c_3 k_i \quad (22)$$

where w_i is the weight of node i , d_m is the number of POIs of category m (emergency, recreation, education, etc.) located around node i within a radius of 100 m, P_i is the total number of passengers accessing node i , k_i is the node degree, c_1 , c_2 , and c_3 are scaling factors. Certain POIs which are equidistant to multiple stops are allocated to the nearest node with the least distance. Fig. 6 shows the heat map indicating the nodes serving high demand in

Hong Kong. In a comparison between the nodes serving high demand areas and the nodes with high centrality values, we notice about 60% similarity of the nodes being compared, indicating that nodes of high topological centrality are also serving relatively higher demand areas. However, the remaining 40% nodes, though are topologically central, are serving low demand areas. This shows that merely considering topological features but ignoring their actual usage might lead to unrealistic conclusions, and such information is important information to operators to carefully design and optimize the network. Fig. 7 shows the comparison of highly central nodes versus nodes serving high demand areas. Thus, the demand estimation method would address the practical usage of topologically central nodes.

V. Notable Contributions to Public Transports Network Analysis

In this section, we discuss a few notable contributions in the field of PTN analysis in addition to the applications of network metrics in the study of PTN topologies.

- i) The usual procedure for generating the topology of a PTN is based on some available online dataset. Kurant and Thiran [15] made a novel attempt to extract real physical topology of a network by considering the time-tables of the mass transportation systems. Despite the different terminologies adopted (space-of-changes for P-space representation, space-of-stations for L-space representation and the other being space-of-stops representation), the representations proposed by Kurant and Thiran [15] are generally consistent with the representation types discussed in Section III. Essentially, a multilayer framework had been adopted considering the actual mapping of logical graphs on physical graphs, where the

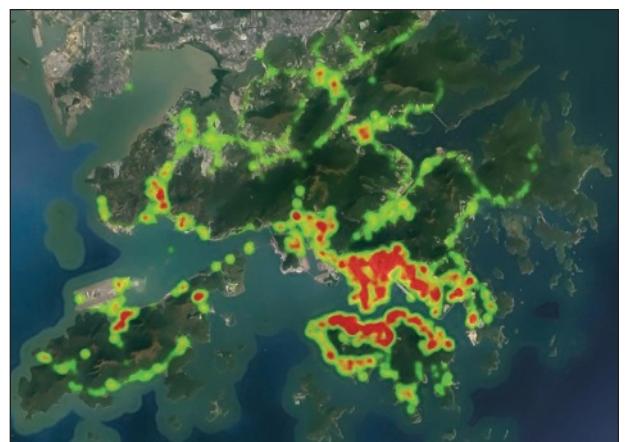


Figure 6. Heat map showing the nodes serving higher demand areas (red) in Hong Kong.

logical layer is the real-world traffic flow layer and the physical layer is the topological representation based on space-of-changes, space-of-stations and space-of-stops. A node load was estimated based on the weighted combination of four load estimators, namely, node degree, betweenness, restricted betweenness, and simple load (origin-destination pair), assuming the combined estimation would aid in revealing some hidden network information which only degraded the performance of the best involved estimator (simple load). Moreover, Kurant and Thiran [15] also acknowledged the fact that only the OD-pair information would not suffice to carry out node load estimation without additional information like the traffic pattern.

- ii) A rare but insightful attempt was made by Haznagy *et al.* [12] to apply the page ranking concept in a PTN analysis. The public transport stops are ranked, in a

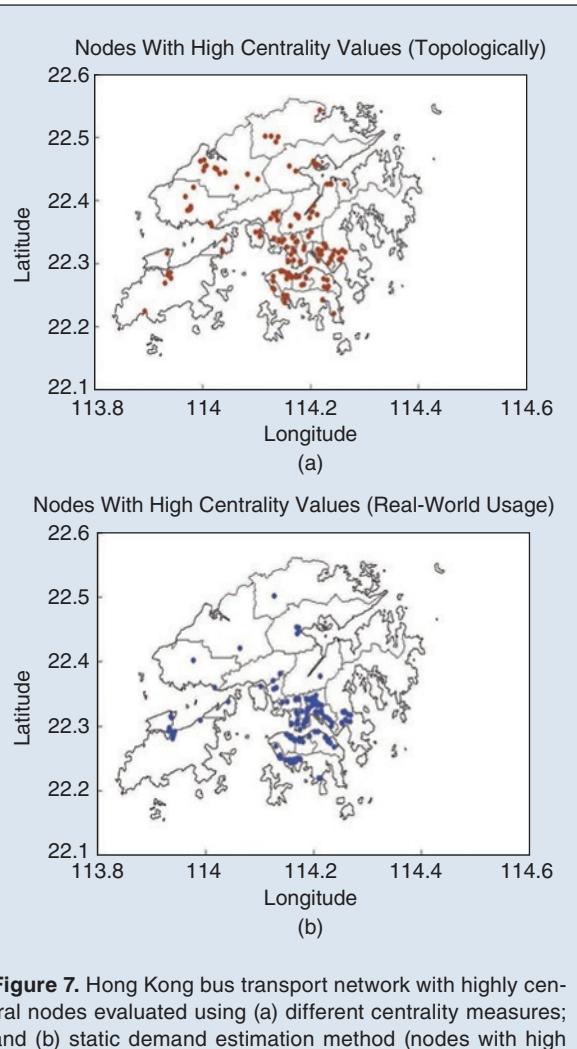


Figure 7. Hong Kong bus transport network with highly central nodes evaluated using (a) different centrality measures; and (b) static demand estimation method (nodes with high centrality or high node weight are the nodes with the normalized value greater than or equal to 0.8.)

similar manner as in web page ranking in a search engine demonstrated earlier by Larry Page [64]. The idea behind evaluating the pagerank is to identify the key nodes in the network that have significant impact in analyzing the transport efficiency.

- iii) Spatial embedding networks (SENs) have been introduced by Yang *et al.* [10] to demonstrate the effectiveness in capturing the topological properties alongside with the underlying spatial characteristics of a network. It has been demonstrated that, considering the underlying geographical feature is as important as considering the network topology in PTN analysis. A concept of extended space (ES) model was adopted to represent the L-space (ESL), P-space (ESP) and networks with SSPs (ESW) representation. A flexible transfer algorithm using the extended model was also proposed to evaluate the cost of a transfer plan (cd) taking into account factors like transfer time, walking distance, and distance to taking buses. Such analysis has practical significance as it provides the passengers a list of top minimum cost transfer path routes.
- iv) A simple network evolution model using a quasi-continuous approximation model was proposed by Chen *et al.* [23]. In their work, the number of bus routes a stop joins, R , and the bus stop's degree, k , are the key parameters. Based primarily on the preferential attachment, a simple BTN model was organized by adding one new route at a time. It was demonstrated empirically that a strong linear correlation exists between R and k , and this formed the basis for the evolution model [23].
- v) A new P-space representation that considers the uplink and downlink routes separately for the bus routes in Harbin (a northeastern Chinese city) was proposed by Feng *et al.* [26]. Essentially, the representation introduced a duplication factor $DF_{R'}$ which is the ratio of repeated stations to unique stations for a given route R' . This parameter provides practical useful information about the bus route's spatial availability, and $DF_{R'}$ was found to exceed 36%. In the new representation, the adjacency matrix element a_{ij} is assigned a value 1 if the node is a part of both uplink and downlink routes, and 0.5 if the node is a part of either uplink or downlink route. This representation readily captures the richness of a node in terms of the degree, weighted degree, average shortest path length, and node weight (weighted degree/degree). The basis for evaluating the richness parameter is the so-called rich-club phenomenon, i.e., the correlation probability of nodes having high richness parameter (hub nodes). An exponential distribution was observed by probing the rich-club

connectivity pattern, indicating that in a small portion of the network, the hub nodes are well connected. Furthermore, the evaluated node weight showed positive correlation with the corresponding degree, weighted degree, and number of routes along a node (R), indicating that the stations carrying maximum load are always well connected [26].

- vi) A simple and realistic routing algorithm called passenger intuitive logic (PIL) was used by Wu *et al.* [32] to study the passenger flow in metro networks. The passengers' intuitive strategy of choosing routes, including minimizing the number of hops traversed and the number of transfers made, formed the basis of the routing algorithm used in the study. In the study, Wu *et al.* combined the use of shortest path (SP) and minimum transfer path (MTP) to determine the routes chosen by passengers. Here, MTP corresponds to the route that has the least number of transfer times, i.e.,

$$P_{\text{MTP}} = \left(1 - \frac{\varepsilon^2}{\lambda_{\text{th}}^2}\right)^{\frac{1}{2}} \left(1 - \frac{(\gamma' - \xi)^2}{\xi^2}\right)$$

where $\varepsilon \in [0, \lambda_{\text{th}}]$, $\gamma' \in [0, \xi]$; and $P_{\text{SP}} = 1 - P_{\text{MTP}}$

(23)

where P_{MTP} is the probability of taking a minimum transfer path, and P_{SP} is the probability of taking a shortest path. Simulation results for the Beijing, Tokyo, Hong Kong, and London metro systems offer insightful observations on the relationship between the topological structure of metro networks and traffic flow [32].

- vii) Topological efficiency of traffic networks has traditionally been evaluated in terms of the number of hops as given by

$$\eta_G = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{d_{ij}}$$
(24)

where d_{ij} is the shortest distance path between nodes i and j . In a recent work [13], we proposed an alternative approach to measure the network efficiency in terms of time rather than distance since the time metric is more naturally used by passengers, i.e.,

$$\eta_{G,t} = \frac{1}{N(N-1)} \sum_{\substack{i=1 \dots n-1 \\ j=i+1 \dots n}} \frac{d_{ij}}{v_{ij}}$$
(25)

where d_{ij} is the total number of hops between nodes i and j , N is the network size, and v_{ij} is the maximum velocity attained along every hop with the shortest path between nodes i and j [13]. Due to the constraint in obtaining real-world data of v_{ij} in (25), we

employed the SUMO (Simulation of Urban Mobility) simulator to validate the time efficiency for a single route using the synthetic mobility trace which yielded a better estimation of time efficiency.

SUMO is a microscopic multi-modal traffic simulator which allows the user to explicitly control the behavior of each vehicle [65]. To conduct the simulation, we first build the road network topology by importing the actual road topology including information on road junctions, bus stops, POIs, and traffic lights according to Openstreetmap [66], as shown in Fig. 8(a). Then, the routes for buses are set up according to the actual time table, and other generic vehicles are set up using Activitygen, an activity-based traffic generator [67]. Finally, results are extracted from the SUMO

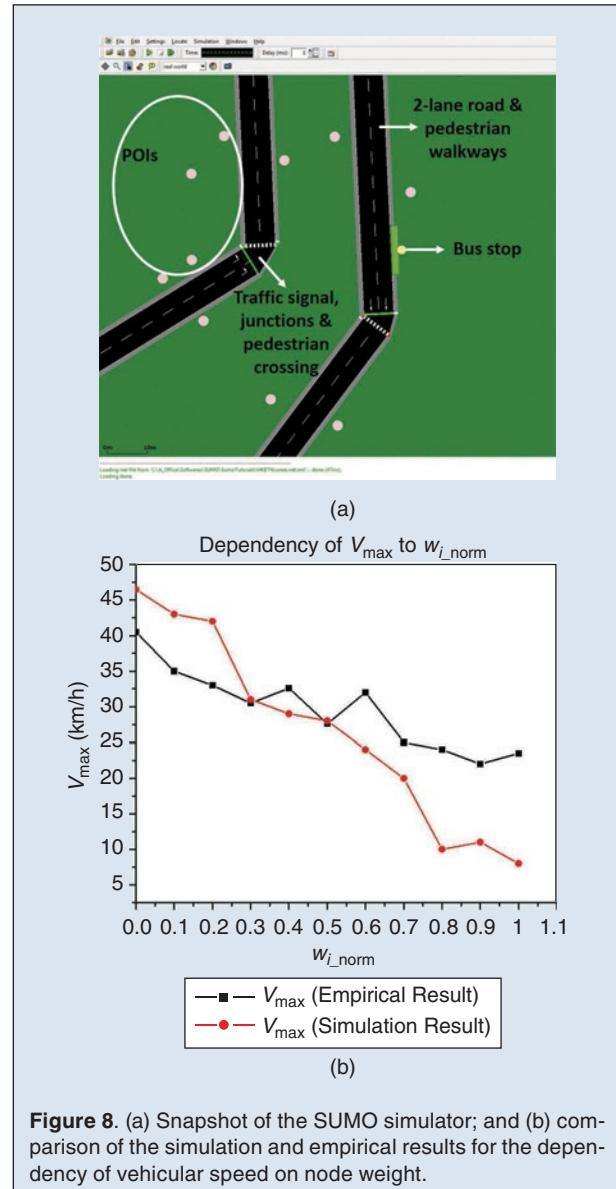


Figure 8. (a) Snapshot of the SUMO simulator; and (b) comparison of the simulation and empirical results for the dependency of vehicular speed on node weight.

Extracting useful information from huge and distributed datasets remains a major challenge. In public transport network analysis, the size of datasets, typically consisting of several thousand nodes, is relatively midget and the time for data mining is also comparatively manageable.

output files which record the footprints of every vehicle during the simulation time at a sampling rate of 1 sec. By evaluating the maximum speed for every road segment and the geographic distance between the stops, the end-to-end travel delay can be calculated using (25).

Our results showed the dependency of the vehicular speed along a road segment upon the node weight, as discussed in Section IV-J. Specifically, we observed that the higher the node weight, the lower the maximum speed attained by the vehicles on the road segment, especially during rush hours. The speed was observed to be further affected when the distance between the stops is reduced. Our simulation results have been verified using real-world data provided by the Kowloon Motor Bus Co., one of the major transport operators in Hong Kong [13]. Fig. 8(b) shows the dependency of the maximum speed attained along a road segment (V_{\max}) for a normalized node weight ($w_{i,\text{norm}}$). Empirical data are in good agreement with our simulation results. We may conclude that with increased node weight (demand) and reduced geographical distance between the stops, the attainable speed by vehicles along a road segment is reduced significantly. In practice, when the bus stops are located closer to each other to offer better service, traffic speed will be compromised, and more aggressive reduction of distance between the stops may even lead to a state of traffic congestion. Our node weight model can be adopted to facilitate a better route planning and stop deployment to maintain optimal traffic performance.

VI. Conclusion and Future Work

In a data driven world, the availability of real-world datasets and high-end tools for handling huge datasets has greatly facilitated the research of complex system and data analysis. Extracting useful information from huge and distributed datasets remains a major challenge. In public transport network (PTN) analysis, the size of datasets, typically consisting of several thousand nodes, is relatively midget and the time for data mining is also comparatively manageable. Despite the successful attempts in applying concepts from network science to PTN analysis, serious study of PTN from a network science perspective is still relatively rare.

In this paper, we aimed at bringing together some of the recent developments in the application of network theory to PTN analysis. In particular, useful contributions have been made by various researchers in the use of L-space representation in comparison to P-, B- and C-space representations, since the L-space graph structure mimics the actual real-world infrastructure of a PTN. A directed and weighted network structure is best suited for the study of bus transport structures, whereas an undirected and weighted network structure is more suited for metro transport studies, and the main reason for considering the graph type is the level of overlapping among inbound and outbound routes. We have found that the notion of supernodes offers practical and more insightful perspective to understanding the actual network behavior, which is difficult to be captured by conventional graph representations. Furthermore, adding static weights to nodes and edges has been found to be effective in capturing the significance of nodes and links in PTNs. It is worth noting that merely representing the PTN structure as a graph and analyzing various network parameters may not lead to practically useful conclusions because the purpose of the public transport systems is to meet travel needs of the community being served, which requires the consideration of more practical network parameters. Also, considering the spatial embedding of PTNs alongside with the topological analysis conveys more insightful information without which quantifying the network might yield rudimentary results.

Topological analysis of PTNs have been performed using various local metrics (e.g., degree, clustering, betweenness centrality, closeness centrality), global metrics (e.g., degree distribution, scale-free property, average path length, small-world property), and pairwise properties (e.g., assortativity and communities). The study of various local, global, and pairwise properties has provided intriguing information about the topological behavior of public transport networks. Such study has provided a great source of information for researchers in the applied fields, for example, in designing of transfer algorithm, optimization of public transport routes, prediction and regulation of road congestion, network planning, transit operation, etc. However, while PTN analysis generates information like the existence of hierarchical structure, core-periphery structure, and the absence of scaling in a PTN, such information does not find immediate practical relevance to the PTN operators or government agencies. Thus, more work

is still needed in developing application-oriented network analysis so that results produced from network theory can be readily translated to useful practical information and more desirably at the operational level.

Robustness analysis is another important area. Evaluating the resilience of PTNs improves understanding of various criteria of network breakdown under different attack strategies. Future research topics may also include the study of the passenger migration process, the application of integrated multiple transport modal analysis to analyze real-world complexity of passenger route selection, effects of polarization of stops and routes on the demand flow in the network, etc. Furthermore, while research efforts have been devoted to the spatial dynamics of PTNs in the past, the temporal dynamics reflecting the topological variation of a PTN at different times of the day should deserve serious attention. Another major area of research is dealing with the integration of the multiple transport networks to form a coordinated and complimentary transport system that can significantly enhance the traffic carrying capacity and efficiency of the entire system. In the past, very little contribution has been made through multi-layer analysis where individual transport networks are treated as independent topologies, and understanding the interaction among these layers should deserve more research attention in view of the practical relevance of integrated PTNs.

Finally, we would like to emphasize that applying graph theory to the analysis of public transport behavior offers an effective and convenient way to understand the network operation at both the local and global levels. The various spaces of network representation provide the fundamental network representation framework for analyzing PTNs. The incorporation of practical network parameters and the emphasis of the dynamic spatio-temporal behavior of the network can offer a broader and more practical view of the network functionality relevant to the network operators. Alongside with offering these advantages, the network-based analysis also raises a few technical challenges as a consequence of increased computational time with increasing network size and the lack of real-world datasets. In closing, we believe that PTN analysis from a graph theory perspective will continue to uncover important network properties and to serve as a solid foundation on which to develop performance optimization strategies, network planning, service deployment, maintenance schedules, etc. for achieving better and more sustainable transport services and eventually smarter cities.

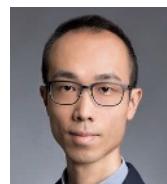
Acknowledgment

We would like to thank Prof. John W. Polak of Imperial College London for his insightful suggestions in organizing this survey, and for shedding light on the areas of

future work. We also thank Xingtang Wu of Beijing Jiaotong University for his contribution towards sharing information on metro network analysis. This work was supported in part by the National Natural Science Foundation of China (Project 61401384) and the Hong Kong Polytechnic University (Project 4-BCCH).



Tanuja Shanmukhappa received the B.E. degree in Electronics and Communication Engineering from the Kuvempu University in 2010, and the M.Tech. degree in Telecommunication Engineering from Visvesvaraya Technological University (VTU) in 2013. She was awarded with the Ph.D. degree in Electronic and Information Engineering from the Hong Kong Polytechnic University, Hong Kong, in 2019. She was the recipient of university gold medal in her master's degree in 2013. Tanuja completed her exchange program at Imperial College London, UK, in 2018. Her current research interests include graph theory and network science, transportation network analysis, and vehicular ad-hoc networks (VANET). Tanuja also had industry experience working as an intern and trainee at Infineon Technologies and NXP semiconductors, India, respectively. She is a student member of IEEE.



Ivan Wang-Hei Ho (M'10–SM'18) received the B.Eng. and M.Phil. degrees in information engineering from The Chinese University of Hong Kong, Hong Kong, in 2004 and 2006, respectively, and the Ph.D. degree in electrical and electronic engineering from the Imperial College London, London, UK, in 2010. He was a research intern with the IBM Thomas J. Watson Research Center, Hawthorne, NY, USA, and a Postdoctoral Research Associate with the System Engineering Initiative, Imperial College London. In 2010, he co-founded P2 Mobile Technologies Ltd, where he served as the Chief Research and Development Engineer. He is currently an Assistant Professor with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. His research interests include wireless communications and networking, specifically in vehicular networks, intelligent transportation systems, the Internet of Things, and complex network analysis of transportation and communication networks. He primarily invented the MeshRanger series wireless mesh embedded system, which received the Silver Award in Best Ubiquitous Networking at the Hong Kong ICT Awards 2012. He is currently an Associate Editor of the IEEE Access and the IEEE Transactions on Circuits and Systems II, and served as the TPC Co-Chair of the CoWPoR Workshop in conjunction with IEEE

SECON 2018 and the PERSIST-IoT Workshop in conjunction with ACM MobiHoc 2019.



Chi K. Tse (M'90–SM'97–F'06) received the B.Eng. (Hons.) degree with first class honors in electrical engineering and the Ph.D. degree from the University of Melbourne, Parkville, Australia, in 1987 and 1991, respectively. He is currently the Chair Professor at the Hong Kong Polytechnic University, Hong Kong, where he was the Head of the Department of Electronic and Information Engineering from 2005 to 2012. He is the author/coauthor of ten books, 20 book chapters, and more than 500 papers in research journals and conference proceedings, and holds five U.S. patents. Dr. Tse received a number of research and industry awards, including the Best Paper Award by the IEEE TRANSACTIONS ON POWER ELECTRONICS in 2001, the Best paper Award by the International Journal of Circuit Theory and Applications in 2003, two Gold Medals at the International Inventions Exhibition in Geneva in 2009 and 2013, and was awarded a number of recognitions by the academic and research communities, including honorary professorship by several Chinese and Australian universities, Chang Jiang Scholar Chair Professorship, IEEE Distinguished Lectureship, Distinguished Research Fellowship by the University of Calgary, Gledden Fellowship, and International Distinguished Professorship-at-Large by the University of Western Australia. While with the Hong Kong Polytechnic University, he received the President's Award for Outstanding Research Performance twice, the Faculty Research Grant Achievement Award twice, the Faculty Best Researcher Award, and several teaching awards. He serves and has served as the Editor-in-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II (2016–2017), the IEEE Circuits and Systems Magazine (2012–2015), and the IEEE Circuits and Systems Society Newsletter (since 2007), an Associate Editor for three IEEE Journal/Transactions, an Editor for the International Journal of Circuit Theory and Applications, and is on the editorial boards of a few other journals. He also serves as a Panel Member of Hong Kong Research Grants Council and NSFC and a Member of several professional and government committees.



Kin K. Leung (M'86–SM'93–F'01) received his B.S. degree from the Chinese University of Hong Kong in 1980, and his M.S. and Ph.D. degrees from University of California, Los Angeles, in 1982 and 1985, respectively. He joined AT&T Bell Labs in New Jersey in 1986 and worked at its successors, AT&T

Labs and Lucent Technologies Bell Labs, until 2004. Since then, he has been the Tanaka Chair Professor in the Electrical and Electronic Engineering (EEE), and Computing Departments at Imperial College in London. He is the Head of Communications and Signal Processing Group in the EEE Department. His current research focuses on protocols, optimization and modeling of wireless networks and computer systems. He also works on multi-antenna and cross-layer designs for wireless networks. He received the Distinguished Member of Technical Staff Award from AT&T Bell Labs in 1994, and was a co-recipient of the Lanchester Prize Honorable Mention Award in 1997. He was elected an IEEE Fellow in 2001, received the Royal Society Wolfson Research Merits Award in 2004–09 and became a member of Academia Europaea in 2012. Along with his co-authors, he received several best paper awards, including the IEEE PIMRC 2012, ICDCS 2013 and ICC 2019. He served as a member (2009–11) and the chairman (2012–15) of the IEEE Fellow Evaluation Committee for ComSoc. He has served as a guest editor for the IEEE JSAC, IEEE Wireless Communications and the MONET journal, and as an editor for the JSAC: Wireless Series, IEEE Trans. on Wireless Communications and IEEE Trans. on Communications. Currently, he is an editor for the ACM Computing Survey and International Journal on Sensor Networks.

References

- [1] M. Kuby, S. Tierney, T. Roberts, and C. Upchurch, "A comparison of geographic information systems, complex networks, and other models for analyzing transportation network topologies," 2005, pp. 1–58.
- [2] A.-L. Barabási, *Linked: The New Science of Networks*. Perseus Books Group, 2003.
- [3] L. Euler, "Leonhard Euler and the Königsberg bridges," *Sci. Amer.*, vol. 189, no. 1, pp. 66–72, 1953.
- [4] K. T. S. Oldham, *The Doctrine of Description: Gustav Kirchhoff, Classical Physics, and the "Purpose of All Science" in 19th Century Germany*. Berkeley, CA: Univ. of California, 2008.
- [5] A.-L. Barabási, *Network Science*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
- [6] M. Newman, *Networks: An Introduction*. London, U.K.: Oxford Univ. Press, 2010.
- [7] M. Van Steen, *Graph Theory and Complex Networks: An Introduction*. VU Amsterdam, 2010.
- [8] J. Lee and D. W. Wong, *Statistical Analysis with ArcView GIS*. New York: Wiley, 2001.
- [9] H. J. Miller and S.-L. Shaw, *Geographic Information Systems for Transportation: Principles and Applications*. London, U.K.: Oxford Univ. Press, 2001.
- [10] X.-H. Yang, G. Chen, S.-Y. Chen, W.-L. Wang, and L. Wang, "Study on some bus transport networks in China with considering spatial characteristics," *Transp. Res. A, Policy Practice*, vol. 69, pp. 1–10, 2014.
- [11] A. Chatterjee, M. Manohar, and G. Ramadurai, "Statistical analysis of bus networks in India," *PloS One*, vol. 11, no. 12, pp. 1–17, 2016.
- [12] A. Háznyag, I. Fi, A. London, and T. Németh, "Complex network analysis of public transportation networks: A comprehensive study," in *Proc. IEEE 2015 Int. Conf. Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2015, pp. 371–378.
- [13] T. Shanmukhappa, I. W.-H. Ho, and C. K. Tse, "Spatial analysis of bus transport networks using network theory," *Phys. A, Stat. Mech. Appl.*, vol. 502, pp. 295–314, 2018.
- [14] G. A. Pavlopoulos, D. Paez-Espino, N. C. Kyriopoulos, and I. Iliopoulos, "Empirical comparison of visualization tools for larger-scale network analysis," *Adv. Bioinf.*, vol. 2017, 2017.

- [15] M. Kurant and P. Thiran, "Extraction and analysis of traffic and topologies of transportation networks," *Phys. Rev. E*, vol. 74, no. 3, pp. 1–10, 2006.
- [16] J. Sienkiewicz and J. A. Hołyst, "Statistical analysis of 22 public transport networks in Poland," *Phys. Rev. E*, vol. 72, no. 4, pp. 1–11, 2005.
- [17] C. Von Ferber, T. Holovatch, Y. Holovatch, and V. Palchykov, "Public transport networks: Empirical analysis and modeling," *Eur. Phys. J. B*, vol. 68, no. 2, pp. 261–275, 2009.
- [18] H. Zhang, P. Zhao, J. Gao, and X.-m. Yao, "The analysis of the properties of bus network topology in Beijing basing on complex networks," *Math. Problems Eng.*, vol. 2013, 2013.
- [19] X. Zhang, G. Chen, Y. Han, and M. Gao, "Modeling and analysis of bus weighted complex network in Qingdao city based on dynamic travel time," *Multimedia Tools Appl.*, vol. 75, no. 24, pp. 17,553–17,572, 2016.
- [20] X. Xu, J. Hu, F. Liu, and L. Liu, "Scaling and correlations in three bus-transport networks of China," *Phys. A, Stat. Mech. Appl.*, vol. 374, no. 1, pp. 441–448, 2007.
- [21] H. Zhang, "Structural analysis of bus networks using indicators of graph theory and complex network theory," *Open Civil Eng. J.*, vol. 11, no. 1, 2017.
- [22] A. Chatterjee, "Studies on the structure and dynamics of urban bus networks in Indian cities," arXiv Preprint, arXiv:1512.05909, 2015.
- [23] Y.-Z. Chen, N. Li, and D.-R. He, "A study on some urban bus transport networks," *Phys. A, Stat. Mech. Appl.*, vol. 376, pp. 747–754, 2007.
- [24] X. Qing, Z. Zhenghu, X. Zhijing, W. Zhang, and T. Zheng, "Space P-based empirical research on public transport complex networks in 330 cities of China," *J. Transp. Syst. Eng. Inf. Technol.*, vol. 13, no. 1, pp. 193–198, 2013.
- [25] A. De Bona, K. Fonseca, M. Rosa, R. Lüders, and M. Delgado, "Analysis of public bus transportation of a Brazilian city based on the theory of complex networks using the P-space," *Math. Problems Eng.*, vol. 2016, 2016.
- [26] S. Feng, B. Hu, C. Nie, and X. Shen, "Empirical study on a directed and weighted bus transport network in China," *Phys. A, Stat. Mech. Appl.*, vol. 441, pp. 85–92, 2016.
- [27] L. Sun, Y. Lu, and D.-H. Lee, "Understanding the structure of urban bus networks: The C-space representation approach," in *Proc. COTA Int. Conf. Transportation Professionals*, 2015, pp. 1557–1567.
- [28] Y. Yang, Y. Liu, M. Zhou, F. Li, and C. Sun, "Robustness assessment of urban rail transit based on complex network theory: A case study of the Beijing subway," *Safety Sci.*, vol. 79, pp. 149–162, 2015.
- [29] J. Zhang, M. Zhao, H. Liu, and X. Xu, "Networked characteristics of the urban rail transit networks," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 6, pp. 1538–1546, 2013.
- [30] S. Derrible, "Network centrality of metro systems," *PloS One*, vol. 7, no. 7, pp. 1–10, 2012.
- [31] J. Zhang, X. Xu, L. Hong, S. Wang, and Q. Fei, "Networked analysis of the Shanghai subway network, in China," *Phys. A, Stat. Mech. Appl.*, vol. 390, no. 23–24, pp. 4562–4570, 2011.
- [32] X. Wu, H. Dong, C. K. Tse, I. W. Ho, and F. C. Lau, "Analysis of metro network performance from a complex network perspective," *Phys. A, Stat. Mech. Appl.*, vol. 492, pp. 553–563, 2018.
- [33] J. Feng, X. Li, B. Mao, Q. Xu, and Y. Bai, "Weighted complex network analysis of the Beijing subway system: Train and passenger flows," *Phys. A, Stat. Mech. Appl.*, vol. 474, pp. 213–223, 2017.
- [34] S. Derrible and C. Kennedy, "The complexity and robustness of metro networks," *Phys. A, Stat. Mech. Appl.*, vol. 389, no. 17, pp. 3678–3691, 2010.
- [35] K. Lee, W.-S. Jung, J. S. Park, and M. Choi, "Statistical analysis of the metropolitan Seoul subway system: Network structure and passenger flows," *Phys. A, Stat. Mech. Appl.*, vol. 387, no. 24, pp. 6231–6234, 2008.
- [36] P. Angeloudis and D. Fisk, "Large subway systems as complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 367, pp. 553–558, 2006.
- [37] Y. Shi, H. Lu, C. Nie, and H. Yoshitsugu, "Analysis of public transport networks in Nagoya based on complex network theory," in *Logistics: The Emerging Frontiers of Transportation and Development in China*, 2009, pp. 2759–2766.
- [38] P. Erdős and A. Rényi, "On random graphs," *Publ. Math. (Debrecen)*, vol. 6, pp. 290–294, 1959.
- [39] P. Erdős and A. Rényi, "On the evolution of random graphs," *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, vol. 5, pp. 17–61, 1960.
- [40] A.-L. Barabási and E. Bonabeau, "Scale-free networks," *Sci. Amer.*, vol. 288, no. 5, pp. 60–69, 2003.
- [41] A.-L. Barabási, "Scale-free networks: A decade and beyond," *Science*, vol. 325, no. 5939, pp. 412–413, 2009.
- [42] P.-P. Zhang et al., "Model and empirical study on some collaboration networks," *Phys. A, Stat. Mech. Appl.*, vol. 360, no. 2, pp. 599–616, 2006.
- [43] G. Fagiolo, "Clustering in complex directed networks," *Phys. Rev. E*, vol. 76, no. 2, pp. 1–8, 2007.
- [44] C. Von Ferber, Y. Holovatch, and V. Palchykov, "Scaling in public transport networks," arXiv Preprint, arXiv:cond-mat/0501296, 2005.
- [45] W. Contributors, "Shortest path problem — Wikipedia, the free encyclopedia," 2018.
- [46] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [47] F. Ivis, "Calculating geographic distance: Concepts and methods," in *Proc. 19th Conf. Northeast SAS User Group*, 2006.
- [48] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [49] M. D. Humphries and K. Gurney, "Network 'small-world-ness': A quantitative method for determining canonical network equivalence," *PloS One*, vol. 3, no. 4, pp. 1–10, 2008.
- [50] Q. K. Telesford, K. E. Joyce, S. Hayasaka, J. H. Burdette, and P. J. Laurienti, "The ubiquity of small-world networks," *Brain Connectivity*, vol. 1, no. 5, pp. 367–375, 2011.
- [51] J. Travers and S. Milgram, "The small world problem," *Psychol. Today*, vol. 1, no. 1, pp. 61–67, 1967.
- [52] M. Barthélémy, "Spatial networks," *Phys. Rep.*, vol. 499, no. 1–3, pp. 1–101, 2011.
- [53] T. Shanmukhappa, I. W. H. Ho, C. K. Tse, X. Wu, and H. Dong, "Multi-layer public transport network analysis," in *Proc. IEEE Int. Symp. Circuits and Systems*, 2018, pp. 1–5.
- [54] M. J. Alenazi and J. P. Sterbenz, "Evaluation and comparison of several graph robustness metrics to improve network resilience," in *Proc. 7th Int. Workshop Reliable Networks Design and Modeling (RNDM)*, 2015, pp. 7–13.
- [55] M. J. Alenazi and J. P. Sterbenz, "Comprehensive comparison and accuracy of graph metrics in predicting network resilience," in *Proc. 11th Int. Conf. Design of Reliable Communication Networks*, 2015, pp. 157–164.
- [56] J. Wang, "Robustness of complex networks with the local protection strategy against cascading failures," *Safety Sci.*, vol. 53, pp. 219–225, 2013.
- [57] J. Wu, M. Barahona, Y.-J. Tan, and H.-Z. Deng, "Spectral measure of structural robustness in complex networks," *IEEE Trans. Syst. Man. Cybern. A, Syst. Hum.*, vol. 41, no. 6, pp. 1244–1252, 2011.
- [58] R. Pastor-Satorras, A. Vázquez, and A. Vespignani, "Dynamical and correlation properties of the internet," *Phys. Rev. Lett.*, vol. 87, no. 25, pp. 1–4, 2001.
- [59] M. E. Newman, "Assortative mixing in networks," *Phys. Rev. Lett.*, vol. 89, no. 20, pp. 208,701, 2002.
- [60] J. G. Foster, D. V. Foster, P. Grassberger, and M. Paczuski, "Edge direction and the structure of networks," *Proc. Nat. Acad. Sci.*, vol. 107, no. 24, pp. 10,815–10,820, 2010.
- [61] M. E. Newman, "Detecting community structure in networks," *Eur. Phys. J. B*, vol. 38, no. 2, pp. 321–330, 2004.
- [62] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, pp. 1–15, 2004.
- [63] B. S. Khan and M. A. Niazi, "Network community detection: A review and visual survey," arXiv Preprint, arXiv:1708.00977, 2017.
- [64] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, Tech. Rep., 1999.
- [65] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo – Simulation of urban mobility: An overview," in *Proc. 3rd Int. Conf. Advances in System Simulation*, ThinkMind, 2011.
- [66] Openstreetmap. Accessed on: Nov 13, 2015. [Online]. Available: <https://www.openstreetmap.org/#map=5/51.500/-0.100/>
- [67] "Activitygen." Accessed on: Nov 13, 2015. [Online]. Available: http://sumo.dlr.de/wiki/Demand/Activity-based_Demand_Generation/
- [68] T. Shanmukhappa, I. W.-H. Ho, and C. K. Tse, "Bus transport network in Hong Kong: Scale-free or not?" in *Proc. Int. Symp. Nonlinear Theory and Its Applications*, 2016, pp. 610–613.
- [69] V. Latora and M. Marchiori, "Is the Boston subway a small-world network?" *Phys. A, Stat. Mech. Appl.*, vol. 314, no. 1–4, pp. 109–113, 2002.

2019 Index

IEEE Circuits and Systems Magazine

Vol. 19

This index covers all technical items—papers, correspondence, reviews, etc.—that appeared in this periodical during 2019, and items from previous years that were commented upon or corrected in 2019. Departments and other items may also be covered if they have been judged to have archival value.

The Author Index contains the primary entry for each item, listed under the first author's name. The primary entry includes the coauthors' names, the title of the paper or other item, and its location, specified by the publication abbreviation, year, month, and inclusive pagination. The Subject Index contains entries describing the item under all appropriate subject headings, plus the first author's name, the publication abbreviation, month, and year, and inclusive pages. Note that the item title is found only under the primary entry in the Author Index.

Author Index

B

Basu, S., and Dewilde, P., [From the Guest Editors]; *CAS-M First Quarter 2019* 7-8

C

Cabric, D., *see* Yan, H., *CAS-M Second Quarter 2019* 33-58
Cabric, D., *see* Hoyhtya, M., *CAS-M Third Quarter 2019* 34-45
Chen, F., *see* Xiang, L., *CAS-M Second Quarter 2019* 8-32
Chen, G., *see* Xiang, L., *CAS-M Second Quarter 2019* 8-32
Chen, P., *see* Van, L., *CAS-M First Quarter 2019* 33-54
Chiumento, A., *see* Hoyhtya, M., *CAS-M Third Quarter 2019* 34-45

D

Dewilde, P., *see* Basu, S., *CAS-M First Quarter 2019* 7-8

F

Forsell, M., *see* Hoyhtya, M., *CAS-M Third Quarter 2019* 34-45

G

Gallagher, T., *see* Yan, H., *CAS-M Second Quarter 2019* 33-58

H

Hoyhtya, M., Mammela, A., Chiumento, A., Pollin, S., Forsell, M., and Cabric, D., Database-Assisted Spectrum Prediction in 5G Networks and Beyond: A Review and Future Challenges; *CAS-M Third Quarter 2019* 34-45

K

Khoo, I., *see* Van, L., *CAS-M First Quarter 2019* 33-54
Koelpin, A., *see* Trautmann, M., *CAS-M Third Quarter 2019* 23-33
Kummert, A., *see* Schwerdtfeger, T., *CAS-M First Quarter 2019* 55-C3

L

Leung, K., *see* Shanmukhappa, T., *CAS-M Fourth Quarter 2019* 39-65
Lian, Y., [President's Message]; *CAS-M First Quarter 2019* 4-8
Ling, C., *see* Yan, H., *CAS-M Second Quarter 2019* 33-58
Lu, J., *see* Wang, X., *CAS-M Third Quarter 2019* 6-22

M

Mammela, A., *see* Hoyhtya, M., *CAS-M Third Quarter 2019* 34-45

Digital Object Identifier 10.1109/MCAS.2019.2945496
Date of current version: 19 November 2019

Mitra, S., *see* Vaidyanathan, P., *CAS-M First Quarter 2019* 14-32

P

Pollin, S., *see* Hoyhtya, M., *CAS-M Third Quarter 2019* 34-45

R

Radakovits, D., *see* TaheriNejad, N., *CAS-M Fourth Quarter 2019* 6-18

Ramesh, S., *see* Yan, H., *CAS-M Second Quarter 2019* 33-58

Reddy, H., *see* Van, L., *CAS-M First Quarter 2019* 33-54

Ren, W., *see* Xiang, L., *CAS-M Second Quarter 2019* 8-32

S

Sanftl, B., *see* Trautmann, M., *CAS-M Third Quarter 2019* 23-33

Schwerdtfeger, T., and Kummert, A., Nonlinear Circuit Simulation by Means of Alfred Fettweis' Wave Digital Principles; *CAS-M First Quarter 2019* 55-C3

Shanmukhappa, T., Wang-Hei Ho, I., Tse, C., and Leung, K., Recent Development in Public Transport Network Analysis From the Complex Network Perspective; *CAS-M Fourth Quarter 2019* 39-65

T

TaheriNejad, N., and Radakovits, D., From Behavioral Design of Memristive Circuits and Systems to Physical Implementations; *CAS-M Fourth Quarter 2019* 6-18

Tognari, R., *see* Zhao, Y., *CAS-M Fourth Quarter 2019* 19-38

Trautmann, M., Sanftl, B., Weigel, R., and Koelpin, A., Simultaneous Inductive Power and Data Transmission System for Smart Applications; *CAS-M Third Quarter 2019* 23-33

Tse, C., *see* Shanmukhappa, T., *CAS-M Fourth Quarter 2019* 39-65

V

Vaidyanathan, P., and Mitra, S., Robust Digital Filter Structures: A Direct Approach; *CAS-M First Quarter 2019* 14-32

Van, L., Khoo, I., Chen, P., and Reddy, H., Symmetry Incorporated Cost-Effective Architectures for Two-Dimensional Digital Filters; *CAS-M First Quarter 2019* 33-54

Vlcek, M., *see* Zahradnik, P., *CAS-M Second Quarter 2019* 59-66

W

Wang, X., and Lu, J., Collective Behaviors Through Social Interactions in Bird Flocks; *CAS-M Third Quarter 2019* 6-22

Wang-Hei Ho, I., *see* Shanmukhappa, T., *CAS-M Fourth Quarter 2019* 39-65

Weigel, R., *see* Trautmann, M., *CAS-M Third Quarter 2019* 23-33

Wu, C., [From the Editor]; *CAS-M First Quarter 2019* 3-8

X

Xia, X., *see* Zhao, Y., *CAS-M Fourth Quarter 2019* 19-38

Xiang, L., Chen, F., Ren, W., and Chen, G., Advances in Network Controllability; *CAS-M Second Quarter 2019* 8-32

Y

Yan, H., Ramesh, S., Gallagher, T., Ling, C., and Cabric, D., Performance, Power, and Area Design Trade-Offs in Millimeter-Wave Transmitter Beamforming Architectures; *CAS-M Second Quarter 2019* 33-58

Z

- Zahradnik, P.**, and Vlcek, M., The World of Ripples; *CAS-M Second Quarter 2019* 59-66
Zhao, Y., Xia, X., and Togneri, R., Applications of Deep Learning to Audio Generation; *CAS-M Fourth Quarter 2019* 19-38

Subject Index**Numeric****5G mobile communication**

- Database-Assisted Spectrum Prediction in 5G Networks and Beyond: A Review and Future Challenges. *Hoyhtya, M.*, +, *CAS-M Third Quarter 2019* 34-45
 Performance, Power, and Area Design Trade-Offs in Millimeter-Wave Transmitter Beamforming Architectures. *Yan, H.*, +, *CAS-M Second Quarter 2019* 33-58

A**Acoustic signal processing**

- Applications of Deep Learning to Audio Generation. *Zhao, Y.*, +, *CAS-M Fourth Quarter 2019* 19-38

Adversarial machine learning

- Applications of Deep Learning to Audio Generation. *Zhao, Y.*, +, *CAS-M Fourth Quarter 2019* 19-38

Approximation theory

- The World of Ripples. *Zahradnik, P.*, +, *CAS-M Second Quarter 2019* 59-66

Array signal processing

- Performance, Power, and Area Design Trade-Offs in Millimeter-Wave Transmitter Beamforming Architectures. *Yan, H.*, +, *CAS-M Second Quarter 2019* 33-58

Audio systems

- Applications of Deep Learning to Audio Generation. *Zhao, Y.*, +, *CAS-M Fourth Quarter 2019* 19-38

Awards

- [Recognitions]. *CAS-M Fourth Quarter 2019* 3, C3
 New IEEE Fellows [Recognitions]. *CAS-M First Quarter 2019* 9

B**Band-pass filters**

- The World of Ripples. *Zahradnik, P.*, +, *CAS-M Second Quarter 2019* 59-66

Behavioral sciences

- Collective Behaviors Through Social Interactions in Bird Flocks. *Wang, X.*, +, *CAS-M Third Quarter 2019* 6-22

- From Behavioral Design of Memristive Circuits and Systems to Physical Implementations. *TaheriNejad, N.*, +, *CAS-M Fourth Quarter 2019* 6-18

Birds

- Collective Behaviors Through Social Interactions in Bird Flocks. *Wang, X.*, +, *CAS-M Third Quarter 2019* 6-22

C**Cellular networks**

- Database-Assisted Spectrum Prediction in 5G Networks and Beyond: A Review and Future Challenges. *Hoyhtya, M.*, +, *CAS-M Third Quarter 2019* 34-45

Chebyshev approximation

- The World of Ripples. *Zahradnik, P.*, +, *CAS-M Second Quarter 2019* 59-66

Circuit simulation

- Nonlinear Circuit Simulation by Means of Alfred Fettweis' Wave Digital Principles. *Schwerdtfeger, T.*, +, *CAS-M First Quarter 2019* 55-C3

Circuit synthesis

- Nonlinear Circuit Simulation by Means of Alfred Fettweis' Wave Digital Principles. *Schwerdtfeger, T.*, +, *CAS-M First Quarter 2019* 55-C3

Complex networks

- Advances in Network Controllability. *Xiang, L.*, +, *CAS-M Second Quarter 2019* 8-32

Complex systems

- Collective Behaviors Through Social Interactions in Bird Flocks. *Wang, X.*, +, *CAS-M Third Quarter 2019* 6-22

- Recent Development in Public Transport Network Analysis From the Complex Network Perspective. *Shanmukhappa, T.*, +, *CAS-M Fourth Quarter 2019* 39-65

Computational complexity

- Database-Assisted Spectrum Prediction in 5G Networks and Beyond: A Review and Future Challenges. *Hoyhtya, M.*, +, *CAS-M Third Quarter 2019* 34-45

Computational modeling

- Database-Assisted Spectrum Prediction in 5G Networks and Beyond: A Review and Future Challenges. *Hoyhtya, M.*, +, *CAS-M Third Quarter 2019* 34-45

Computer architecture

- Performance, Power, and Area Design Trade-Offs in Millimeter-Wave Transmitter Beamforming Architectures. *Yan, H.*, +, *CAS-M Second Quarter 2019* 33-58

- Symmetry Incorporated Cost-Effective Architectures for Two-Dimensional Digital Filters. *Van, L.*, +, *CAS-M First Quarter 2019* 33-54

Computer vision

- Applications of Deep Learning to Audio Generation. *Zhao, Y.*, +, *CAS-M Fourth Quarter 2019* 19-38

Control theory

- Advances in Network Controllability. *Xiang, L.*, +, *CAS-M Second Quarter 2019* 8-32

- Collective Behaviors Through Social Interactions in Bird Flocks. *Wang, X.*, +, *CAS-M Third Quarter 2019* 6-22

Controllability

- Advances in Network Controllability. *Xiang, L.*, +, *CAS-M Second Quarter 2019* 8-32

D**Data models**

- Database-Assisted Spectrum Prediction in 5G Networks and Beyond: A Review and Future Challenges. *Hoyhtya, M.*, +, *CAS-M Third Quarter 2019* 34-45

Data transmission

- Simultaneous Inductive Power and Data Transmission System for Smart Applications. *Trautmann, M.*, +, *CAS-M Third Quarter 2019* 23-33

Deep learning

- Applications of Deep Learning to Audio Generation. *Zhao, Y.*, +, *CAS-M Fourth Quarter 2019* 19-38

Design tools

- From Behavioral Design of Memristive Circuits and Systems to Physical Implementations. *TaheriNejad, N.*, +, *CAS-M Fourth Quarter 2019* 6-18

Digital filters

- Nonlinear Circuit Simulation by Means of Alfred Fettweis' Wave Digital Principles. *Schwerdtfeger, T.*, +, *CAS-M First Quarter 2019* 55-C3

- Robust Digital Filter Structures: A Direct Approach. *Vaidyanathan, P.*, +, *CAS-M First Quarter 2019* 14-32

F**Filtering theory**

- Robust Digital Filter Structures: A Direct Approach. *Vaidyanathan, P.*, +, *CAS-M First Quarter 2019* 14-32

- Symmetry Incorporated Cost-Effective Architectures for Two-Dimensional Digital Filters. *Van, L.*, +, *CAS-M First Quarter 2019* 33-54

- The World of Ripples. *Zahradnik, P.*, +, *CAS-M Second Quarter 2019* 59-66

Finite element analysis

- Nonlinear Circuit Simulation by Means of Alfred Fettweis' Wave Digital Principles. *Schwerdtfeger, T.*, +, *CAS-M First Quarter 2019* 55-C3

Finite impulse response filters

- Robust Digital Filter Structures: A Direct Approach. *Vaidyanathan, P.*, +, *CAS-M First Quarter 2019* 14-32

- Symmetry Incorporated Cost-Effective Architectures for Two-Dimensional Digital Filters. *Van, L.*, +, *CAS-M First Quarter 2019* 33-54

+ Check author entry for coauthors

G

Graph theory

Recent Development in Public Transport Network Analysis From the Complex Network Perspective. Shanmukhappa, T., +, CAS-M Fourth Quarter 2019 39-65

I

IIR filters

Symmetry Incorporated Cost-Effective Architectures for Two-Dimensional Digital Filters. Van, L., +, CAS-M First Quarter 2019 33-54

Inductive power transmission

Simultaneous Inductive Power and Data Transmission System for Smart Applications. Trautmann, M., +, CAS-M Third Quarter 2019 23-33

Integrated circuit modeling

From Behavioral Design of Memristive Circuits and Systems to Physical Implementations. TaheriNejad, N., +, CAS-M Fourth Quarter 2019 6-18

K

Kalman filters

Advances in Network Controllability. Xiang, L., +, CAS-M Second Quarter 2019 8-32

L

Land transportation

Recent Development in Public Transport Network Analysis From the Complex Network Perspective. Shanmukhappa, T., +, CAS-M Fourth Quarter 2019 39-65

Learning systems

Applications of Deep Learning to Audio Generation. Zhao, Y., +, CAS-M Fourth Quarter 2019 19-38

Linear systems

Advances in Network Controllability. Xiang, L., +, CAS-M Second Quarter 2019 8-32

M

Mathematical model

Collective Behaviors Through Social Interactions in Bird Flocks. Wang, X., +, CAS-M Third Quarter 2019 6-22

Memristors

From Behavioral Design of Memristive Circuits and Systems to Physical Implementations. TaheriNejad, N., +, CAS-M Fourth Quarter 2019 6-18

Millimeter wave communication

Performance, Power, and Area Design Trade-Offs in Millimeter-Wave Transmitter Beamforming Architectures. Yan, H., +, CAS-M Second Quarter 2019 33-58

Modeling

Database-Assisted Spectrum Prediction in 5G Networks and Beyond: A Review and Future Challenges. Hoyhtya, M., +, CAS-M Third Quarter 2019 34-45

Multi-agent systems

Collective Behaviors Through Social Interactions in Bird Flocks. Wang, X., +, CAS-M Third Quarter 2019 6-22

N

Network theory (graphs)

Recent Development in Public Transport Network Analysis From the Complex Network Perspective. Shanmukhappa, T., +, CAS-M Fourth Quarter 2019 39-65

Network topology

Recent Development in Public Transport Network Analysis From the Complex Network Perspective. Shanmukhappa, T., +, CAS-M Fourth Quarter 2019 39-65

+ Check author entry for coauthors

Nonlinear circuits

Nonlinear Circuit Simulation by Means of Alfred Fettweis' Wave Digital Principles. Schwerdtfeger, T., +, CAS-M First Quarter 2019 55-C3

P

Passband

Robust Digital Filter Structures: A Direct Approach. Vaidyanathan, P., +, CAS-M First Quarter 2019 14-32

Phase shifters

Performance, Power, and Area Design Trade-Offs in Millimeter-Wave Transmitter Beamforming Architectures. Yan, H., +, CAS-M Second Quarter 2019 33-58

Phased arrays

Performance, Power, and Area Design Trade-Offs in Millimeter-Wave Transmitter Beamforming Architectures. Yan, H., +, CAS-M Second Quarter 2019 33-58

Public transportation

Recent Development in Public Transport Network Analysis From the Complex Network Perspective. Shanmukhappa, T., +, CAS-M Fourth Quarter 2019 39-65

R

Radio frequency

Performance, Power, and Area Design Trade-Offs in Millimeter-Wave Transmitter Beamforming Architectures. Yan, H., +, CAS-M Second Quarter 2019 33-58

RLC circuits

Robust Digital Filter Structures: A Direct Approach. Vaidyanathan, P., +, CAS-M First Quarter 2019 14-32

S

Sensitivity

Robust Digital Filter Structures: A Direct Approach. Vaidyanathan, P., +, CAS-M First Quarter 2019 14-32

Simulation

From Behavioral Design of Memristive Circuits and Systems to Physical Implementations. TaheriNejad, N., +, CAS-M Fourth Quarter 2019 6-18

Smart grids

Simultaneous Inductive Power and Data Transmission System for Smart Applications. Trautmann, M., +, CAS-M Third Quarter 2019 23-33

Speech recognition

Applications of Deep Learning to Audio Generation. Zhao, Y., +, CAS-M Fourth Quarter 2019 19-38

SPICE

Nonlinear Circuit Simulation by Means of Alfred Fettweis' Wave Digital Principles. Schwerdtfeger, T., +, CAS-M First Quarter 2019 55-C3

Spread spectrum communication

Database-Assisted Spectrum Prediction in 5G Networks and Beyond: A Review and Future Challenges. Hoyhtya, M., +, CAS-M Third Quarter 2019 34-45

T

Task analysis

The World of Ripples. Zahradnik, P., +, CAS-M Second Quarter 2019 59-66

Transfer functions

Symmetry Incorporated Cost-Effective Architectures for Two-Dimensional Digital Filters. Van, L., +, CAS-M First Quarter 2019 33-54

Advertising Sales Offices

IEEE CIRCUITS AND SYSTEMS MAGAZINE REPRESENTATIVE

Erik Henson
Naylor Association Solutions
Phone: +1 352 333 3443
Fax: +1 352 331 3525
ehenson@naylor.com

445 Hoes Lane, Piscataway, NJ 08854

Digital Object Identifier 10.1109/MCAS.2018.2880540

March Issue

Ad Closing:

Space Reservations January 15
Ad Material January 17

Recognitions

(continued from page 3)

on Circuits and Systems for Video Technology, Volume: 26, Issue: 7, July 2016.

VLSI Transactions Best Paper Award

Jason Kamran Eshraghian, Jr., Kyoungrok Cho, Ciyan Zheng, Minho Nam, Herbert Ho-Ching Iu, Wen Lei, and Kamran Eshraghian, "Neuromorphic Vision Hybrid RRAM-CMOS Architecture." IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 26, Issue 12, Dec. 2018.

Outstanding Young Author Award

Abhronil Sengupta, et al., "Proposal for an All-Spin Artificial Neural Network: Emulating Neural and Synaptic Functionalities Through Domain Wall Motion in Ferromagnets," IEEE Transactions on Biomedical Circuits and Systems, Volume 10, Issue 6, Dec. 2016.

Pre-Doctoral Scholarship

Oiga Krestinskaya

Chapter of the Year Award

Singapore

Regions 1–7 Chapter of the Year Award

Santa Clara Valley Chapter

Region 8 Chapter of the Year Award

Spain

Region 10 Chapter of the Year Award

Shanghai



What + If = IEEE

420,000+ members in 160 countries.
Embrace the largest, global, technical community.

People Driving Technological Innovation.

ieee.org/membership

#IEEEmember



KNOWLEDGE

COMMUNITY

PROFESSIONAL DEVELOPMENT

CAREER ADVANCEMENT