THE IEEE Intelligent Informatics BULLETIN

IEEE COMPUTER SOCIETY

IEEE Computer Society
Technical Committee
on Intelligent Informatics

December 2018 Vol. 19 No. 2 (ISSN 1727-5997)

Profile

Feature Articles

Announcements

Special Issue on IEEE Intelligent Informatics Bulletin Call For Papers

On-line version: http://www.comp.hkbu.edu.hk/~iib (ISSN 1727-6004)

IEEE Computer Society Technical Committee on Intelligent Informatics (TCII)

The Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society deals with tools and systems using biologically and linguistically motivated computational paradigms such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality. If you are a member of the IEEE Computer Society, you may join the TCII without cost at http://computer.org/tcsignup/.

The IEEE Intelligent Informatics Bulletin

Aims and Scope

The IEEE Intelligent Informatics Bulletin is the official publication of the Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society, which is published once a year in both hardcopies and electronic copies. The contents of the Bulletin include (but may not be limited to):

1) Letters and Communications of the TCII Executive Committee

2) Feature Articles

3) R&D Profiles (R&D organizations, interview profile on individuals, and projects etc.)

4) Book Reviews

5) News, Reports, and Announcements (TCII sponsored or important/related activities)

Materials suitable for publication at the IEEE Intelligent Informatics Bulletin should be sent directly to the Associate Editors of respective sections.

Technical or survey articles are subject to peer reviews, and their scope may include the theories, methods, tools, techniques, systems, and experiences for/in developing and applying biologically and linguistically motivated computational paradigms, such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, and self-organization in the research and application domains, such as data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality.

Publisher: *The IEEE Computer Society Technical Committee on Intelligent Informatics*
Address: *Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (Attention: Dr. William K. Cheung; Email:william@comp.hkbu.edu.hk)*
ISSN Number: *1727-5997(printed)1727-6004(on-line)*
Abstracting and Indexing: *All the published articles will be submitted to the following on-line search engines and bibliographies databases for indexing—Google(www.google.com), The ResearchIndex(citeseer.nj.nec.com), The Collection of Computer Science Bibliographies (liinwww.ira.uka.de/bibliography/index.html), and* ***DBLP*** *Computer Science Bibliography (www.informatik.uni-trier.de/~ley/db/index.html).*

Coordinated, Multimodal Neuromodulation and Neuroimaging

PCNC (Psychology Clinical Neuroscience Center) and MRN (MIND Research Network)

I. Context

This profile spans two research centers. Dr. Vincent P. Clark is the founding Director of the Psychology Clinical Neuroscience Center (PCNC), which is located within the Department of Psychology at the University of New Mexico. The mission of the PCNC is the development of new knowledge regarding how normal and abnormal behavior and cognition arise from the brain.

Dr. Clark was formerly Scientific Director of the MIND Research Network (MRN), which is located on UNM's campus, and is a non-profit organization that is a member of the Lovelace Family of companies. It offers a combination of methods for neuroimaging, and focuses on a combination of imaging, data analysis and bioinformatics for clinical and cognitive neuroscience research.

II. Description

Faculty associated with the PCNC and MRN have a long history of performing structural and functional imaging studies of the human nervous system, using a variety of neuroimaging modalities, including electroencephalography (EEG), magnetoencephalography (MEG), functional magnetic resonance imaging (fMRI) and others. Each method offers a variety of advantages and disadvantages for a particular neuroscience question, depending on their spatial and temporal resolution, relative cost and signal contrast to noise at the individual and group levels. By combining imaging methods, additional information is gained, and can be useful to confirm findings across modalities.

A. Applications of Neuroimaging

These neuroimaging methods have been applied to a variety of questions. Our work has included both the brain basis of healthy human cognition, primarily perception, attention and memory, as well as differences in brain organization associated with psychiatric illnesses such as addiction and schizophrenia, and neurological disorders such as traumatic brain injury, stroke and Huntington's disease, among others. We have found that imaging can predict relapse in recovery from stimulant addiction with up to 80% accuracy [1] (Figure 1). Through the efforts of our laboratories and others, neuroimaging has made great strides in understanding the brain basis of healthy cognition as well as neurological and psychiatric illness.

Non-invasive brain stimulation combined with neuroimaging has the potential to revolutionize cognitive enhancement and provide more effective treatments for brain and mental illness

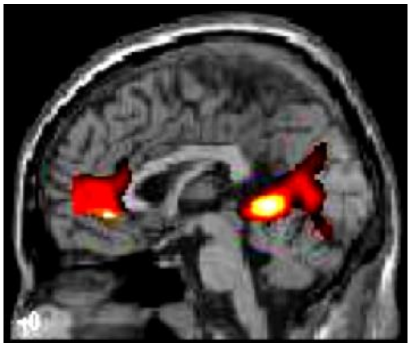


Figure 1. Brain regions whose reduced magnitude of activity predicts relapse in recovering stimulant addicts using fMRI, from [1].

B. Limitations of Neuroimaging

While there has been tremendous impact of neuroimaging in gaining a better understanding of the brain basis of cognition and illness in academic circles, the impact of this work on real-world problems and goals has been limited. One is that neuroimaging is mostly observational, providing correlations between clinical or behavioral variables and brain states. As with any correlation, these methods are unable to test causation directly, which must be verified using other means. In addition, while neuroimaging has been successful in the diagnosis of neurological illness, and to a lesser extent psychiatric illnesses, it has been a failure both in terms of reducing the impact of neurological and mental illness on human health. This is shown in part by the increasing relative rank order of major brain and mental illnesses compared with other forms of illness [2]. Within the disability-adjusted life-year loss ranks for the top 30 diseases and injuries in the US from 1990 and 2010, the most prevalent brain and mental illnesses (ranked from more to less impact of lost life-years in 2010: major depression, drug use disorders, Alzheimer's disease, anxiety disorders, alcohol use disorder and schizophrenia) have moved up in their relative ranking on the average, while physical ailments like road injury and HIV/AIDS have reduced greatly over the same period. This suggests that treatments for brain and mental illness have not improved as quickly as other forms of medicine over the past decades.

Without effective methods of treatment for these disorders, more accurate diagnoses obtained from neuroimaging do not provide substantial benefits. In order to apply information derived from neuroimaging and take full advantage of its anatomical and temporal characterizations of brain function, new methods of treatment must be developed in order to use this information to its fullest possible benefit. Being able to normalize brain function, that is, to guide disordered and pathological brain function to a healthier state, may be one key to reducing the impact of or curing brain and mental illness.

C. Non-invasive Brain Stimulation (NIBS)

Over a decade ago, our laboratory began experimenting with NIBS in conjunction with neuroimaging. All techniques for NIBS apply some form of energy to the nervous system, including electromagnetism (electricity, magnetism and light) and physical pressure (ultrasound and others) in order to influence its activity, and thus to influence behaviour. This includes short-term changes during treatment, and longer term effects related to neuroplasticity. Because NIBS can be applied to targeted anatomical locations with a specific temporal profile, it can be used to influence neural activity supporting particular behaviours or symptoms. Also, when used within pre-defined limits of energy deposition, it is found to be safe, in part because of its anatomical focus that reduces effects on other body organs.

While NIBS offers hope for improved methods of treatment, it has been plagued by a number of failed replications [3] and uncertainty regarding its mechanisms of action, or indeed if any real changes in brain function other than placebo effects occur under certain situations. In these cases, neuroimaging may provide a benefit to NIBS, by documenting changes in brain function associated with its application, and by using neuroimaging to guide the application of NIBS to achieve a specific change in brain function with greater efficacy and reliability. The following describes some methods of NIBS used in our laboratory, and efforts in our laboratory to combine NIBS with neuroimaging to optimize and characterize the effects of NIBS.

1) Transcranial Electrical Stimulation (tES)

Electrical brain stimulation takes many forms. The minimum requirements are a controlled current with at least two electrodes closing a circuit across the scalp. Dosages for transcranial direct current stimulation (tDCS) are typically 1-2 mA for 10-30 minutes, although higher current or longer durations and have been used [4]. The number of tDCS protocols that can be reasonably conceived of, including differences in electrodes (size, number and locations) and applied current (polarity, amplitude and duration), results in over 4 million possible combinations. In addition, if methods of current modulation are considered, such as transcranial alternating current stimulation (tACS) where current is modulated in a sine wave fashion, or random noise, pulsed, sawtooth or many other time varying patterns, and also their additive combination, then a nearly infinite variety of protocols are possible. Each protocol may augment or interfere with anatomically distinct sets of neurons operating with different temporal profiles through resonant mechanisms that have been only partially explicated so far, and therefore may have a large variety of different effects.

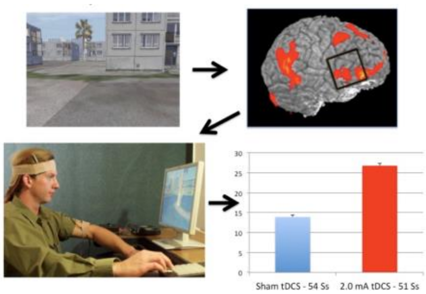


Figure 2. Shows procedures used in [5] to develop a tDCS protocol able to increase learning by a factor of 2. Upper left, an example target detection training stimulus. Upper right, brain regions found using fMRI involved in learning to detect targets. Lower left, application of tDCS during training, positioned over regions indicated. Lower right, effect of verum tDCS (red) vs. sham control (blue) on performance improvement with training.

Our initial studies used a tDCS protocol planned using anatomical data derived from fMRI and MEG studies [5]. Subjects learned to detect targets hidden in complex pictures taken from a virtual training environment. This tDCS protocol approximately doubled learning rate and d', a measure of signal detection, and is one of the largest reported effects of any treatment on learning thus far. This line of research has been replicated in multiple subsequent studies, both in our laboratory at UNM [6] and independently at another university [7]. We have also identified some cognitive [8] and neural mechanisms of this tDCS protocol using fMRI and magnetic resonance spectroscopy [9] and other imaging methods, and are finding inc current studies that other forms of learning are also accelerated using this tDCS protocol.

With our partners at HRL Laboratories, LLC, we have developed a novel method of closed-loop alternating current stimulation (CL-tACS) patterned using EEG recorded during sleep. CL-tACS stimulation was configured to have the same frequency and phase as participants' endogenous slow wave oscillations (0.5 to 1.2 Hz) derived from EEG recorded as they slept. CL-tACS was applied for 5 full cycles at their endogenous frequency and phase over bilateral frontal electrodes (F3 and F4) at 1.5 mA per hemisphere with temporal/mastoid return electrodes. Using this, we found evidence for increased memory consolidation during sleep [10] using the same target learning task used in [5], with a peak effect occurring with about 220 stimulation events [11], and with additional beneficial effects on sleep quality and efficiency [12].

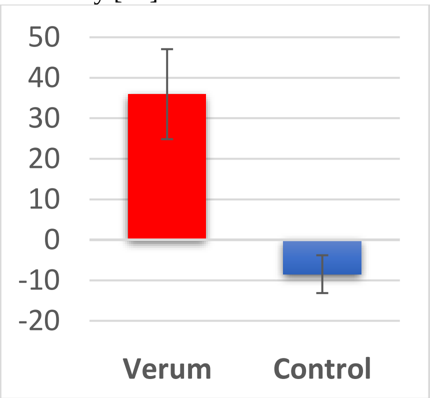


Figure 3. Shows percentage change in MEP amplitude vs. baseline 1 minute after active for verum tUS (red) and sham control tUS (blue).

2) Transcranial Ultrasound Stimulation (tUS)

We have been examining the use of tUS for NIBS. TUS offers the capability of modulating brain regions with higher resolution and anatomical specificity than other methods, and also deeper structures with minimal impact on more superficial areas. Low-intensity ultrasound has been used for imaging, opening the possibility that the identical system may be used for both neuroimaging and neurostimulation. To test this, we recently completed a study using a diagnostic ultrasound system

(CX-50, Philips) built for performing ultrasonic imaging. When used in the imaging HGen, B-mode with harmonics on and a focal depth of 10 cm, we found that 2 minutes of stimulation to motor cortex resulted in approximately 6-10 minutes of increased cortical excitation, as evidenced by increased amplitude of transcranial magnetic stimulation (TMS) induced motor evoked potentials (MEPs) recorded from the hand [13] (Figure 3).

3) Transcranial Light Stimulation (tLS)

TLS uses infrared light frequencies that are able to penetrate the scalp and skull, and that modulate neuronal activity and brain function. We have used a system that transmits near-infrared light through the scalp and nasal tissue, with 820 nm light pulsed at 40 Hz for 20 minutes (Gamma Neuro, Vielight). In pilot studies this protocol has been found to accelerate learning and improve performance using the same target learning task as described above, and with a similar effect size to that obtained using tDCS [5-7]. Larger studies are currently being performed to confirm this result, and to use EEG and other neuroimaging modalities to characterize the tLS induced changes brain function that underlie these changes in behavior.

D. Future Prospects Combining NIBS with Neuroimaging

NIBS combined with neuroimaging offers the hope of developing new methods for cognitive enhancement, and new treatments for brain and mental illness that are less expensive and safer than current standards of care. Neuroimaging offers NIBS a procedure to choose among the nearly infinite variety of protocols that are available, and also to better understand the mechanisms by which NIBS works. In return, NIBS may offer a better method to take full advantage of the anatomical and temporal characterizations of healthy cognition contrasted with neurological and psychiatric illnesses obtained using neuroimaging. Together, they may provide new methods for reducing the suffering caused by neurological and psychiatric disorders. Work in our laboratory and others has provided early examples of the successful application of this combined method. Further work may lead to the discovery of new treatments that are able to reduce the impact of brain and mental illness.

REFERENCES

[1] V.P. Clark, G. Beatty, R.E., Anderson, P. Kodituwakku, J. Phillips, T.D.R. Lane, K.A. Kiehl, V.D. Calhoun. Reduced fMRI activity predicts relapse in patients recovering from stimulant dependence. *Human Brain Mapping*, 35(2), 414-428, 2014.

[2] US Burden of Disease Collaborators. The State of US Health, 1990-2010: Burden of Diseases, Injuries, and Risk Factors. *JAMA*; 310(6): 591–608, 2013.

[3] J.C. Horvath, J.D., Forte, O. Carter Quantitative review finds no evidence of cognitive effects in healthy populations from single-session transcranial direct current stimulation (tDCS). *Brain Stimulation* 8(3):535-550, 2015.

[4] *Practical Guide to Transcranial Direct Current Stimulation: Principles, Procedures and Applications.* H. Knotova, M.A. Nitsche, M. Bikson and A.J. Woods, Editors. Springer, 1st Edition, 2019.

[5] V.P. Clark, B.A. Coffman, A.R. Mayer, M.P. Weisend, T.D.R. Lane, V.D. Calhoun, E.M. Raybourn, C.M. Garcia, E.M. Wassermann. TDCS guided using fMRI significantly accelerates learning to identify concealed objects. *NeuroImage*, 59(1):117-128, 2012.

[6] B.A. Coffman, M.C. Trumbo, R.A. Flores, C.M. Garcia, A.J. van der Merwe, E.M. Wassermann, M.P. Weisend, V.P. Clark. Impact of tDCS on performance and learning of target detection: Interaction with stimulus characteristics and experimental design. *Neuropsychologia*, 50(7):1594-1602, 2012.

[7] B. Falcone, B.A. Coffman, V.P. Clark, R. Parasuraman. Transcranial direct current stimulation augments perceptual sensitivity and 24-hour retention in a complex threat detection task. *PLoS ONE*, 7(4): e34993, 2012.

[8] B.A. Coffman, M.C. Trumbo, V.P. Clark. Enhancement of object detection with transcranial direct current stimulation is associated with increased attention. *BMC Neuroscience*, 13:108, 2012.

[9] M.A. Hunter, B.A. Coffman, C. Gasparovic, V.D. Calhoun, M.C. Trumbo, V.P. Clark. Baseline effects of transcranial direct current stimulation on glutamatergic neurotransmission and large-scale network connectivity. *Brain Research*, 1594:92-107, 2015.

[10] N. Ketz, A.P. Jones, N.B. Bryant V.P. Clark, P.K. Pilly. Closed-loop slow-wave tACS improves sleep dependent long-term memory generalization by modulating endogenous oscillations. *Journal of Neuroscience*, 38(33):7314-7326, 2018.

[11] A.P. Jones, J. Choe, N.B. Bryant, C.S.H. Robinson, N.A. Ketz, S.W. Skorheim, A. Combs, M.L. Lamphere, B. Robert, H.A. Gill, M.D. Heinrich, M.D. Howard, V.P. Clark, P.K. Pilly. Closed-loop tACS delivered during slow-wave sleep enhances consolidation of generalized information. *Frontiers in Neuroscience*, in press, 2018.

[12] C.S.H. Robinson CSH, N.B. Bryant NB, Maxwell JW, Jones AP, Robert B, Lamphere M, Combs A, Azzawi HA, Gibson BC, Sanguinetti JL, Ketz NA, Pilly PK, Clark VP. The benefits of closed-loop transcranial alternating current stimulation on subjective sleep quality. *Brain Sciences*, 8(12):204, 2018.

[13] B.C. Gibson, J.L. Sanguinetti, B.W. Badran, A.B. Yu, E.P. Klein, C.C. Abbott, J.T. Hansberger, V.P. Clark. Increased excitability induced in the primary motor cortex by transcranial ultrasound stimulation. *Frontiers in Neurology*, in press, 2018.

Contact Information

Name: Prof. Vincent P. Clark
Address: MSC03-2220, Dept. Psychology, University of New Mexico, Albuquerque, NM 87131 USA
Email :vclark@unm.edu
Phone: (505) 277-2223
Fax: (505) 277-1394
Website: pcnc.unm.edu

Blockchain Data Analytics

Cuneyt Gurcan Akcora, Matthew F. Dixon, Yulia R. Gel, and Murat Kantarcioglu

***Abstract*—Many novel applications, ranging from cryptocurrencies to food supply chain management, drive consumer and industrial adoption of Blockchain technologies. As these applications proliferate, so does the complexity and volume of data stored by Blockchains. Analyzing this data has emerged as an important research topic, already leading to methodological advancements in the information sciences. In this invited paper, we provide a brief overview of Blockchain Data Analytics, focusing both on the emerging research challenges and on the novel applications – from Bitcoin price prediction to e-crime detection.**

***Index Terms*—Blockchain, Bitcoin, Ethereum, Financial Analytics, Anomaly Detection, Time Series Analysis, Blockchain Data Analytics.**

I. Introduction

THIS decade has been marked with the rise of Blockchain based technologies. At its core, Blockchain is a distributed public ledger that stores transactions between two parties without requiring a trusted central authority. On a Blockchain, two unacquainted parties can create an unmodifiable transaction that is permanently recorded on the ledger to be seen by the public. As legendary venture capitalist Marc Andreessen states “the consequences of this breakthrough are hard to overstate” [1].

The first application of Blockchain has been the Bitcoin [2] cryptocurrency. Bitcoin’s success has ushered an age known as the Blockchain 1.0 [3]. Currently there are more than 1000 Blockchain based cryptocurrencies, known as **alt-coins**. These developments have ignited public interest in Blockchain technology. Some observers compare the inception of Blockchain to the invention of double entry accounting that revolutionized the business world [4]. The emerging Blockchain based applications include voting (FollowMyVote, Social Krona), identity services (Bitnation, Hypr), provenance (Everledger, Chronicled) and copyright management (LBRY, Blockphase). Although it is hard to predict the future impact of Blockchain, it is safe to say that it will enable many important and diverse applications.

Private blockchains are created by industry/organizations and only allow write and read access to the participants with necessary permissions. In contrast, public blockchains, such as Bitcoin, allow any node to join the network without permission and all transactions can be observed by all the nodes that are part of the Blockchain network. In this article we restrict our attention to public blockchains, where data is publicly accessible.

Each blockchain solution utilizes a chain structure, but may also employ novel data structures. Clearly, this information may be analyzed to provide novel insights about emerging trends. This raises questions such as: *1) How to represent and model the data stored on blockchains 2) What are the novel analytical tools needed for analyzing Blockchain data? 3) What insights could be gleaned from the transactions stored on public blockchains?*

We address the above questions by offering a short introduction to Blockchain analytics. We first provide a brief history of public blockchains. Then we discuss the common Blockchain data structure models and provide some insights into several important analytical methods and tools. Finally, we briefly discuss recent studies that use Blockchain data analytics for cryptocurrency modeling, detection of e-crime, human trafficking and illicit economic activity.

II. History

Blockchain was devised and outlined by Satoshi Nakamoto in his “Bitcoin electronic cash system” [2] white paper in 2008.

Although Nakamoto mentioned “a chain of blocks” only, the term Blockchain has become the name of the technology underlying Bitcoin. The success of Bitcoin led to the creation of hundreds of similar cash systems, which came to be known as cryptocurrencies. These off-shoots differ from Bitcoin in a few aspects. For example, Litecoin modified the block mining algorithm for fairness in mining, and ZCash introduced a shielded pool to hide transactions for better privacy.

Although their success is still a hotly debated topic, cryptocurrencies paved the way for broad Blockchain adoption. Since 2014, Blockchain 2.0 led to the creation of Blockchain platforms where software code, called Smart Contracts, can be stored and executed on a Blockchain publicly. These contracts allow unstoppable, unmodifiable and publicly verifiable code execution as transactions between online entities. Blockchain 3.0 is expected to further immerse the technology into our daily lives with IoT integration [5].

Blockchain continues to evolve, but its applications have already matured to rival, and already in some cases, replace more traditional institutions as avenues of global activity. For example, the Ethereum blockchain has become a major fundraising medium for tech start-ups; initial coin offerings (ICOs) of Ethereum tokens have reached 45% of second quarter IPOs [6] in the US.

III. Blockchain Data Models

Public blockchains can be broadly categorized as unspent transaction output (UTXO) based (e.g., Bitcoin, Litecoin) and account based (e.g., Ethereum) blockchains. In both types

The authors are from University of Texas at Dallas Computer Science and Mathematical Sciences departments and Illinois Institute of Technology Applied Mathematics department. Corresponding author e-mail: (cuneyt.akcora@utdallas.edu).

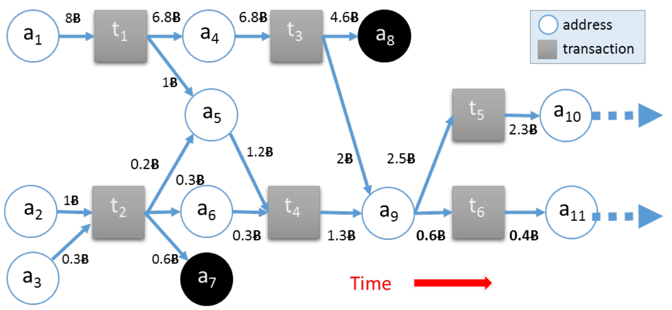


Fig. 1. A network of 11 addresses and 6 Bitcoin transactions. Block boundaries are not shown. Coins at addresses a_7 and a_8 remain unspent. The difference between input and output amounts (e.g., 0.2B at t_1) are collected as the transaction fee. Most crypto-currencies have the same data model as Bitcoin.

of blockchains, a data block consists of a finite number of transactions, but the transactions have differing characteristics. Below, we briefly discuss these two different type of blockchain transaction data.

A. The Unspent Transaction Output Based Blockchain Data

The unspent transaction output (UTXO) based blockchains are the earliest and most valuable (in terms of market capitalization) blockchains: Bitcoin alone constitutes 45-60% of the total cryptocurrency [7] market capitalization. In UTXO blockchains each data block contains a (financial) transaction that encodes a transfer of coins between multiple parties. Each transaction consumes (i.e., spends coins from) some inputs and creates (i.e., directs coins to) new outputs. Fig. 1 shows an example UTXO network, where transaction t_1 encodes a transfer of bitcoins from the address a_1 to a_4 and a_5. In UTXO, coin supply is tied to block creation; a certain amount of coins are created and given to the block miner as the block reward. Bitcoin started with 50B per coin and halves the block reward every 4 years. This geometric series will result in a total of 21 million bitcoins.

We emphasize **three rules** that shape data on UTXO blockchains. These rules are due to the design choices by Satoshi Nakamoto in Bitcoin [2].

Source Rule: Input coins from multiple transactions can be merged and spent in a single transaction (e.g., the address a_5 receives coins from t_1 and t_2 to spent in t_4 in Fig. 1), or spent separately (e.g., in Figure 1, a_9 spends coins received from t_3 and t_4 in t_5 and t_6).

Mapping Rule: Each coin payment must show proof of funds by referencing a set of previous outputs. Although this allows anyone to trace back a history of payments, it is not always possible to locate where a specific coin originates from. This is because each transaction lists a set of inputs and outputs, separately. For example, t_2 has two inputs and three outputs, but an explicit mapping between inputs and outputs does not exist. Coins flowing to a_5 may have come from either a_2 and a_3, or both. As a result, a transaction can be considered a lake with in-flowing rivers, and out-flowing rivers (i.e., emissaries).

Balance Rule: Coins received from one transaction must all be spent in a single transaction. Any amount that is not sent to an output address is considered to be the transaction fee, and gets collected by the miner who creates the block. In order to keep the change, the coin spender can create a new address (i.e., change address) and send the remaining balance to this new address. Another option is to use the spender's address as one of the output addresses, and re-direct the balance. As a community practice, this reuse of the spender's address (i.e., **address reuse**) is discouraged. As a result, most nodes appear in the graph two times only; once when they receive coins and once when they spend it. The change address, if created, becomes the new address of the coin owner.

Due to these rules, the unspent transaction output based blockchains should be considered as forward branching trees, rather than networks.

UTXO blockchains also contain non-transactional data. In the first Bitcoin block, Nakamoto had left the text message "The Times 3 January 2009 Chancellor on brink of second bailout for banks". Adding metadata to Bitcoin transactions have been a topic of discussion and since 2014, each Bitcoin transaction contains a field (*OP_RETURN*) that is designed to store log information in 80 bytes [8].

Improving on the metadata functionality, The Namecoin blockchain has been created in 2011 to store key-value pairs for a decentralized namespace. Namecoin data blocks store registrations or updates for the .bit domain names, which are independent of the ICANN. A domain expires 35,999 blocks (200-250 days) after it is registered as a key-value pair in the Blockchain. Besides the domain registration (i.e., /d), Namecoin has a public online identity namespace (i.e., /id), along with other proposed services.

Storing the full Blockchain to reach .bit domain addresses in real time has discouraged Namecoin adoption. Although online explorer sites and browser extensions have been created to help Internet users with .bit domains, Namecoin namespaces have historically remained underutilized, and most blocks are empty of any key-value pair [9].

B. Account Based Blockchain Data

In account based blockchains, an address can spend a fraction of its coins and keep the remaining balance. In these blockchains, a transaction has exactly one input and one output address. Although address creation is free, mostly a single address is used to receive and send coins multiple times.

Created in 2015, Ethereum [10] is currently the most valuable account based blockchain. Similar to Bitcoin, Ethereum has a currency: Ether. However, the Ethereum project's main goal is to store data and software code on a Blockchain. The code (a smart contract) is written in the proprietary coding language Solidity, which is compiled to bytecode and executed on the Ethereum Virtual Machine. Smart Contracts are self-executing Turing complete contracts which contain code and agreements. An analogy is the MYSQL snippets stored on a database. However, Smart Contracts also ensure unstoppable, deterministic code execution that can be verified publicly.

Account based blockchains use two types of addresses; externally owned addresses (governed by users) and contract addresses (governed by smart contract code). A transaction to upload the Smart Contract code to a contract address is usually initiated by an externally owned address (i.e., user address), but it can also be initiated by a contract address. The code at the address is stored in the Blockchain and replicated at all Blockchain nodes. In other words, uploading the contract forces other nodes to store the code locally.

Similar to the log field in UTXO blockchains, each Ethereum transaction contains an *input data* field which is used to pass messages (i.e., function names and parameters) to smart contracts. The code is executed by feeding parameters to the stored function. This execution occurs at all nodes, worldwide. For this reason, Ethereum is called the World Computer.

Contract creation is expensive, but born by the contract creator. Subsequently, other users or contracts can create transactions directed to the contract address to call the functions contained in the contract. Costs of operations (such as multiplication = 5 and addition = 3) executed by the contract are summed up in terms of the execution fee called “gas”, and billed to the address that created the transaction, in ethers. The currency ether acts as the digital oil of Ethereum World Computer.

Smart Contracts gave rise to Smart Contract based tokens: exchanged data units. Holding tokens allow users to get serviced by a company in real life. For example, the Storj token stores files on your hard disk, and pays you a fee through Ethereum. Furthermore, tokens can be bought or sold online and act as value stores. In this worldwide market, tokens that are valuable are arbitrated in fiat currencies. These prices can be viewed on online exchanges such as coinmarketcap.com

Account based blockchains have two types of transactions. The first transaction type involves a transfer of the used cryptocurrency, such as Ether on Ethereum, between two addresses. This can be modeled with a directed edge between the two addresses.

The second type, internal transactions, are created when smart contracts change states associated with addresses. In the most basic scenario, consider a sell order issued by address a_1 to a Smart Contract where the *to* parameter is a_2 and the *value* parameter is 2 token. The Smart Contract creates an internal transaction that transfers 2 tokens from a_1 to a_2. Internal transactions can be discovered in two ways: by parsing the transaction’s message and updating states associated with a_1 and a_2 manually, or by running the transaction message through the smart contract code and observing the states and logs created during execution. The second option requires running a full Ethereum node and executing every contract transaction, which is costly in terms of time and resources. The parsing option is easier as it does not require code execution. However, the parsing method cannot discover transaction failures (due to reasons such as insufficient gas), and create internal transactions that do not actually exist.

IV. Blockchain Data Analytics Methods and Tools

Largely deriving from existing network analysis methodology, early research works analyze UTXO data by creating a graph that employs a single type of node only. These are transaction and address graph approaches.

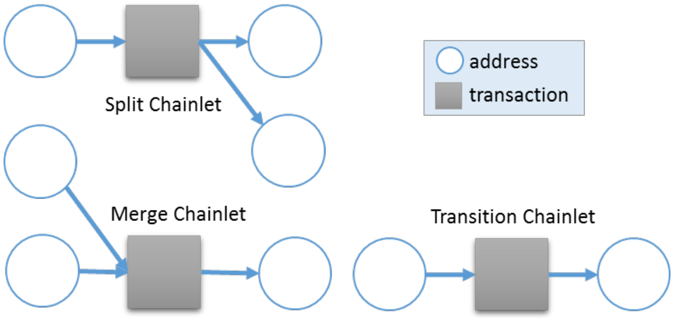


Fig. 2. Chainlets encode a transaction with its inputs and outputs. Chainlets can be aggregated and their occurrence information can then be used in machine learning tasks.

In the transaction graph approach, addresses are ignored and edges are created among transaction nodes [11], [12]. Naturally, the transaction graph is acyclic and a transaction node cannot have new edges in the future.

In the address graph approach [13], transactions are ignored and edges are created among address nodes. However, because of the Mapping Rule (see Sec. III), inputs of a transactions must be connected to all output addresses of a transaction, which may create large cliques if too many addresses are involved in a transaction.

Single node type approaches do not provide a faithful representation of the Blockchain data (See [14] for more on Blockchain data models). The loss of information about addresses or transactions seem to have an impact in predictive models [15].

K-chainlets [16] offer a lossless way to encode network subgraphs where nodes can be addresses or chainlets. The model utilizes local higher order structures of the Blockchain graph. Rather than using individual edges or nodes, subgraphs can be used as the building block in Blockchain analysis. The term *chainlet* refers to such subgraphs.

This choice is due to two reasons. First, the subgraph can be taken as a single data unit because inclusion of nodes and edges in it is based on a single decision. As a transaction is immutable, joint inclusion of input/output nodes in its subgraph cannot be changed afterwards. This is unlike the case on a social network where nodes can become closer on the graph because of actions of their neighbors. Second, as shown in Fig. 2, subgraphs have distinct shapes that reflect their role in the network, and these roles can be aggregated to analyze network dynamics.

As Bitcoin became popular, a number of studies aimed at using various network characteristics for price predictions. For instance, [17], [15] employ such network features as mean account balance, number of new edges and clustering coefficients. In turn, network flows and temporal behavior of the network have been used as alternative price predictors by [18] and [19], respectively.

Studies in network features show that since 2010 the Bitcoin network can be considered a scale-free network [20]. In- and out-degree distributions of the transactions graphs are highly heterogeneous and exhibit a disassortative behavior [21]. Active entities on the network change frequently, but there are consistently active entities [22]. The most central nodes in the network are coin exchange sites [23].

As all transactions are one-to-one, account based blockchains enable the usage of traditional graph analysis tools easily [24], [25]. However care must be taken to extract internal transactions from ordinary transactions, so that all relationships (i.e., token buy/sell) between addresses can be modeled on the graph.

As a second issue, the complete Ethereum graph have overlapping layers of token networks; each token can be represented with a separate graph on the Ethereum network where nodes are user/contract addresses. A token network is a directed, weighted multi-graph. Two token networks may share nodes but not edges. The complete Ethereum graph consists of layers of token networks. Multiple edges can exist between two nodes of the Ethereum graph, and each edge can transfer a different token. On the Ethereum blockchain, it is not rare to see hundreds of edges between two nodes.

A. *Tools*

A criticism of Blockchain is that data blocks are written into files (e.g., as levelDB files in Ethereum and .dat files in Bitcoin) on disk, which makes data querying time consuming. Recent years have witnessed development of Blockchain query languages [26] and analytics frameworks [27] but their adoption is still limited. Companies such as Santiment.com and Chainalysis.com have developed in-house data querying and analysis tools, but these are not yet open to public. Online explorers such as blockchain.com and etherscan.io provide limited analytics tools to the public.

A widely used tool in Bitcoin data analytics is the BlockSci project [28]. A similar tool is the Bitcoin Network Visual Analytics tool Biva.[1]

Besides transaction data that involve financial relations between addresses, the advent of Ethereum 2.0, which brought software code to blockchains, has propelled smart contract analysis [29] as an important data analytics direction. However, most research approaches in this direction are based on static code analysis for tasks such as contract classification [30], and do not analyze the decisions made by the studied smart contracts.

V. Applications of Blockchain Data Analytics

Since the seminal Bitcoin paper [2] in 2008, cryptocurrencies [7] have been the most prominent Blockchain application. Recently there has been an interest in analyzing Blockchain platform (e.g., Ethereum [25]) data but Bitcoin and a few other alt-coins have been the main focus of Blockchain Data Analytics. Broadly, studies address the capacity and limitations for coins to provide a robust and transparent economic system for all economic participants.

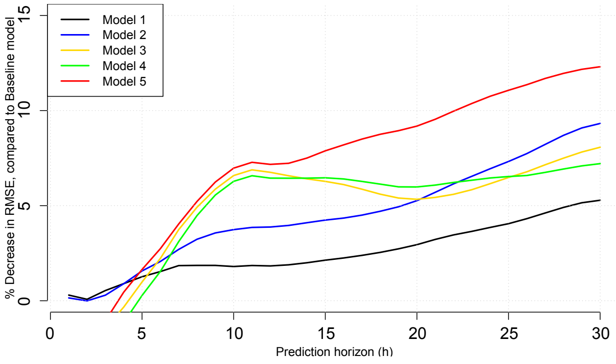


Fig. 3. Daily Bitcoin price prediction in 2016. Model performances for various chainlet models are shown. For three or more steps ahead forecasts, chainlets play an increasingly significant predictive role in Bitcoin price formation, even when other more conventional factors, such as historical price and number of transactions, are accounted for in the model.

a) Price Prediction: One central question is how Bitcoin fares as a financial asset class - in particular whether the transaction graph is linked to price formation and impact liquidity, or even a market crash. Analyzing the relationship between transactions and addresses and Bitcoin price, has therefore become an important analytics research direction [31]. In particular, there is a growing focus on building statistical models which can predict and attribute price movements to transactions and transaction graph properties. While simple Blockchain transactional features, such as average transaction amount, are shown to exhibit mixed performance for cryptocurrency price forecasting [15], a number of recent studies show the utility of global graph features to predict the price [32], [33], [19], [15]. For instance, [17] analyzed the predictive effects of average balance, clustering coefficient, and number of new edges on the Bitcoin price and [16] use Blockchain chainlets as predictors. Two network flow measures were recently proposed by [18] to quantify the dynamics of the Bitcoin transaction network and to assess the relationship between flow complexity and Bitcoin market variables.

The extent to which we can build predictive models from the chainlets has already led to some promising results [16]. In particular, we have been able to identify certain types and groups of chainlets that exhibit predictive influence on Bitcoin price and volatility. Fig. 3 shows the percent decrease in root mean squared error (RMSE) for some of these models relative to a simple baseline model, which uses Bitcoin prices and transaction volumes of previous days only [16]. Specifically, we evaluate $\psi_{M_i}(h)/\psi_{M_0}(h)$, $i = 1, \ldots, 5$, for $h = 1, \ldots, 30$ days ahead, where $\psi_{M_i}(h)$ and $\psi_{M_0}(h)$ are the RMSEs for the chainlet predictive model M_i and the baseline model M_0, respectively. We find that Model 5, which uses five different chainlets, yields the most competitive predictive performance.

In addition to price prediction, chainlets are a lead indicator for price risk - the relative daily loss distribution conditioned on the 'extreme chainlets' leads to more accurate outlier prediction [32]. Without extreme chainlets, risk models underestimate the extreme Bitcoin losses. These extreme chainlets represent transactions from a large number of accounts to fewer addresses or vice versa. Such transactions represent

[1] https://github.com/feog/Biva

systemic movements of coins to and from exchanges and other funds.

Which Blockchain representation? The core idea behind the aforementioned predictive analytics approaches is to extract certain global network features made available through Blockchain and employ them for predictions, with utility in financial markets. The best approach to represent the network is application driven and an open area of research, ranging from approaches which offer a defensible economic interpretation to purely attractive on theoretical grounds. However, there is mounting empirical evidence that Blockchain data augments conventional predictive studies (i.e., combining other data sources with Blockchain data is necessary).

b) Criminal Usage Detection: Since its early days, Bitcoin has been used in dark markets, such as SilkRoad.com to connect illegal vendors with buyers. By design, cryptocurrencies are pseudo-anonymous because users do not need identify themselves to enter the network, but all of their transactions on the Blockchain are public. Knowing this aspect, criminals devise schemes to separate their real life and online identities. Such schemes include connecting to the Blockchain network through privacy-enhancing distributed platforms such as Tor [34]. Furthermore, criminals aim at making their actions indistinguishable from the actions of ordinary users on the Blockchain. This involves creating transactions that look *normal* in terms of frequency, time and amount, with varying success.

Beyond online trade, cryptocurrencies are used in payments for human trafficking [35], ransomware [36], personal blackmails [37] and money laundering [38], among many others. Blockchain Data Analytics tools and algorithms can be used by law agencies [39] to detect and analyze such criminal activities.

Securities governance concerns have arisen around the stability of digital coins as a currency, its susceptibility to price manipulation and illegal usage for money laundering and blackmailing [38], [40], [41], [22], [42]. A key question is the extent to which Blockchain delivers the anonymity which financial criminals seek. The lack of explicit mapping (see the Mapping rule in Sec. III) in UTXO based cryptocurrencies, such as Litecoin and Bitcoin, hinders tracking flow of coins among addresses in time. Although some heuristics [43] have been used to track coins, it is possible to use a series of mixing [44] transactions to hide coin flows. Researchers have found empirical clues for this mixing behavior in the Bitcoin blockchain [45], [13]. Later crypto-currencies such as Zcash [46] and Monero [47] improve the mixing capability by introducing additional measures such as shielded pools. These cryptocurrencies give strong guarantees for anonymity.

The anonymity question can be addressed by first understanding the patterns of criminal usage. Moser et al. [38] analyzed the opportunities and limitations of anti-money laundering (AML) on Bitcoin by identifying how successive transactions are used to transfer money. Blockchain addresses can then be linked to identify suspects behind suspicious transaction patterns in cryptocurrencies [13]. The pattern is usually defined as a repeating transaction involving the movement of digital coins from a black (i.e., affiliated with criminal gains) address to an online exchange, where the coins can be cashed out without being confiscated by authorities. The black address that starts the transaction chain may be related to money laundering [38] and ransomware payment [36]. There is growing evidence on the existence of these illicit activities in Blockchain networks and the reader is referred to [48], [49].

In general, the relative scarcity of wallet addresses labelled as either malicious, fraudulent or the target of ransomware, motivates the application of unsupervised learning. For example, a few known ransom addresses can be used to discover other associated wallet addresses by observing the ’co-spending’ behavior [43]. Other techniques such as oversampling, adaptive penalization and Bayesian networks have been used to address the class imbalance problem in detection of Bitcoin Ponzi schemes [50].

VI. Conclusion

Blockchain technology has recently witnessed a spark of consumer and industrial interest in a broad range of applications, from digital finance to food safety to health care to weapon tracking. As increasingly more new Blockchain applications appear everyday, the complexity and volume of the data stored by Blockchain also rapidly expand – thereby, constituting a new standalone research direction of *Blockchain Data Analytics*.[2] In this invited paper, we provide a brief overview of the current state of Blockchain Data Analytics, focusing both on methodological advancement and the emerging research challenges, as well as offering insight into some of the most important financial applications.

Acknowledgment

The work of Gel is partially supported by NSF IIS 1633331, NSF DMS 1736368 and NSF ECCS 1824710.

The work of Kantarcioglu was supported in part by NIH award 1R01HG006844, NSF awards CICI-1547324, IIS-1633331, CNS-1837627, OAC-1828467 and ARO award W911NF-17-1-0356.

References

[1] A. Marc, “Why bitcoins matters: https://dealbook.nytimes.com/2014/01/21/why-bitcoin-matters/,” *New York Times*, vol. 21, 2014.

[2] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[3] M. Swan, *Blockchain: Blueprint for a new economy*. O’Reilly Media, Inc., 2015.

[4] P. Vigna and M. J. Casey, *The age of cryptocurrency: how bitcoin and the blockchain are challenging the global economic order*. Macmillan, 2016.

[5] S. C. Alliance, “Smart contracts: 12 use cases for business & beyond,” 2016, http://digitalchamber.org/assets/smart-contracts-12-use-cases-for-business-and-beyond.pdf.

[6] C. Long, “Icos were 45% of ipos in q2 2018, as cryptos disrupt investment banks,” www.forbes.com/sites/caitlinlong/2018/07/22/icos-were-45-of-ipos-in-q2-2018-as-cryptos-disrupt-investment-banks.

[7] F. Tschorsch and B. Scheuermann, “Bitcoin and beyond: A technical survey on decentralized digital currencies,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[8] M. Bartoletti and L. Pompianu, “An analysis of bitcoin op_return metadata,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 218–230.

[2] BlockchainTutorial.Github.io

[9] H. A. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan, "An empirical study of namecoin and lessons for decentralized namespace design." in *WEIS*. Citeseer, 2015.

[10] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[11] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis," *arXiv preprint arXiv:1502.01657*, 2015.

[12] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *International Conference on Financial Cryptography and Data Security*. Springer, 2013, pp. 6–24.

[13] M. Spagnuolo, F. Maggi, and S. Zanero, "Bitiodine: Extracting intelligence from the bitcoin network," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 457–468.

[14] C. G. Akcora, Y. R. Gel, and M. Kantarcioglu, "Blockchain: A graph primer," *arXiv preprint arXiv:1708.08749*, pp. 1–17, 2017.

[15] A. Greaves and B. Au, "Using the bitcoin transaction graph to predict the price of bitcoin," *No Data*, 2015.

[16] C. G. Akcora, A. K. Dey, Y. R. Gel, and M. Kantarcioglu, "Forecasting bitcoin price with graph chainlets," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PaKDD)*. Springer, 2018, pp. 765–776.

[17] M. Sorgente and C. Cibils, "The reaction of a network: Exploring the relationship between the bitcoin network structure and the bitcoin price," *No Data*, 2014.

[18] S. Y. Yang and J. Kim, "Bitcoin market return and volatility forecasting using transaction network flow properties," in *IEEE SSCI*, 2015, pp. 1778–1785.

[19] D. Kondor, I. Csabai, J. Szüle, and G. Pósfai, M.and Vattay, "Inferring the interplay between network structure and market effects in bitcoin," *New J. of Phys.*, vol. 16, no. 12, p. 125003, 2014.

[20] M. Lischke and B. Fabian, "Analyzing the bitcoin network: The first four years," *Future Internet*, vol. 8, no. 1, p. 7, 2016.

[21] D. Kondor, M. Pósfai, I. Csabai, and G. Vattay, "Do the rich get richer? an empirical analysis of the bitcoin transaction network," *PloS one*, vol. 9, no. 2, p. e86197, 2014.

[22] M. Ober, S. Katzenbeisser, and K. Hamacher, "Structure and anonymity of the bitcoin transaction graph," *Future internet*, vol. 5, no. 2, pp. 237–250, 2013.

[23] A. Baumann, B. Fabian, and M. Lischke, "Exploring the bitcoin network." in *WEBIST (1)*, 2014, pp. 369–374.

[24] T. Chen, Y. Zhu, Z. Li, J. Chen, X. Li, X. Luo, X. Lin, and X. Zhange, "Understanding ethereum via graph analysis," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1484–1492.

[25] W. Chan and A. Olmsted, "Ethereum transaction graph analysis," in *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE, 2017, pp. 498–500.

[26] S. Bragagnolo, H. Rocha, M. Denker, and S. Ducasse, "Ethereum query language," in *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*. ACM, 2018, pp. 1–8.

[27] M. Bartoletti, S. Lande, L. Pompianu, and A. Bracciali, "A general framework for blockchain analytics," in *Proceedings of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*. ACM, 2017, p. 7.

[28] H. Kalodner, S. Goldfeder, A. Chator, M. Möser, and A. Narayanan, "Blocksci: Design and applications of a blockchain analysis platform," *arXiv preprint arXiv:1709.02489*, 2017.

[29] S. Ducasse, H. Rocha, S. Bragagnolo, M. Denker, and C. Francomme, "Smartanvil: Open-source tool suite for smart contract analysis," 2019.

[30] M. Bartoletti and L. Pompianu, "An empirical analysis of smart contracts: platforms, applications, and design patterns," in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 494–509.

[31] P. Tasca, A. Hayes, and S. Liu, "The evolution of the bitcoin economy: extracting and analyzing the network of payment relationships," *The Journal of Risk Finance*, vol. 19, no. 2, pp. 94–126, 2018.

[32] C. G. Akcora, M. Dixon, Y. R. Gel, and M. Kantarcioglu, "Bitcoin risk modeling with blockchain graphs," *Economics Letters*, pp. 1–5, 2018.

[33] S. Madan, I.and Saluja and A. Zhao, "Automated bitcoin trading via machine learning algorithms," 2015.

[34] P. Syverson, R. Dingledine, and N. Mathewson, "Tor: The secondgeneration onion router," in *Usenix Security*, 2004.

[35] R. S. Portnoff, D. Y. Huang, P. Doerfler, S. Afroz, and D. McCoy, "Backpage and bitcoin: Uncovering human traffickers," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1595–1604.

[36] D. Y. Huang, D. McCoy, M. M. Aliapoulios, V. G. Li, L. Invernizzi, E. Bursztein, K. McRoberts, J. Levin, K. Levchenko, and A. C. Snoeren, "Tracking ransomware end-to-end," in *Tracking Ransomware End-to-end*. IEEE, 2018, pp. 1–12.

[37] A. D. S. Phetsouvanh, F. Oggier, "Egret: Extortion graph exploration techniques in the bitcoin network," in *IEEE ICDM Workshop on Data Mining in Networks (DaMNet)*, 2018.

[38] R. Moser, M.and Bohme and D. Breuker, "An inquiry into money laundering tools in the bitcoin ecosystem," in *eCrime Researchers Summit*. IEEE, 2013, pp. 1–14.

[39] E. U. A. for Law Enforcement Cooperation, "Internet organised crime threat assessment (iocta): https://www.europol.europa.eu/activities-services/main-reports/internet-organised-crime-threat-assessment-iocta-2017," pp. 1–80, 2017.

[40] G. Di Battista, V. Di Donato, M. Patrignani, M. Pizzonia, V. Roselli, and R. Tamassia, "Bitconeview: visualization of flows in the bitcoin transaction graph," in *IEEE VizSec*, 2015, pp. 1–8.

[41] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *IFCA*. Springer, 2013, pp. 34–51.

[42] J. M. Griffin and A. Shams, "Is bitcoin really un-tethered?" 2018.

[43] S. Meiklejohn, M. Pomarole, G. Jordan, D. Levchenko, K.and McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 127–140.

[44] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "Coinshuffle: Practical decentralized coin mixing for bitcoin," in *European Symposium on Research in Computer Security*. Springer, 2014, pp. 345–364.

[45] D. McGinn, D.and Birch, D. Akroyd, M. Molina-Solana, Y. Guo, and W. J. Knottenbelt, "Visualizing dynamic bitcoin transaction patterns," *Big data*, vol. 4, no. 2, pp. 109–119, 2016.

[46] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn, "An empirical analysis of anonymity in zcash," *arXiv preprint arXiv:1805.03180*, 2018.

[47] A. Kumar, C. Fischer, S. Tople, and P. Saxena, "A traceability analysis of moneros blockchain," in *European Symposium on Research in Computer Security*. Springer, 2017, pp. 153–173.

[48] A. Bogner, "Seeing is understanding: anomaly detection in blockchains with visualized features," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 2017, pp. 5–8.

[49] D. D. F. Maesa, A. Marino, and L. Ricci, "Detecting artificial behaviours in the bitcoin users graph," *Online Social Networks and Media*, vol. 3, pp. 63–74, 2017.

[50] M. Bartoletti, B. Pes, and S. Serusi, "Data mining for detecting bitcoin ponzi schemes," *arXiv preprint arXiv:1803.00646*, 2018.

Knowledge Discovery from Temporal Social Networks

Shazia Tabassum, Fabíola S. F. Pereira, and João Gama

***Abstract*—Extracting knowledge from network data is a complex task. It requires the use of appropriate tools and techniques, especially in scenarios that take into account the volume and evolving aspects of the network. There is a vast literature on how to collect, process, and model social media data in the form of networks, as well as key metrics of centrality. However, there is still much to be discussed in relation to the analysis of the underlying huge networks. In this article the goal is to discuss briefly different techniques in the process of gaining knowledge from networked data, especially considering time perspective. Firstly, we presented some techniques for online sampling of temporally ordered massive networked data which can be efficiently plugged in for a further network mining task. Next are discussed approximate mechanisms for high speed network change detection using centrality measures. Hand in hand we also presented, the ways of processing temporal network streams, applications with real data and illustrations using network visualizations. In the end are discussed concepts related to temporal ordering of links and paths in temporal networks.**

***Index Terms*—Temporal Networks, Sampling Evolving Networks, Change Detection, Streaming Network Analysis.**

I. Introduction

Handling and processing, high-velocity networked data generating from real-world applications is a current exigency. Dynamic and evolutionary learning with space and time efficient techniques is an imperative solution for these flooding data networks. We are here focusing this issue in the arena of dynamic sampling and change detection from temporally evolving large networks.

Dynamic Sampling is an exemplary way to deal with the issues relating to massive evolving data, like answering approximate queries, running simulations, understanding and modeling true network structure, inadequate data, detecting events/changes in the network, etc. Apart from other applications one of its major appeals lies in estimating the true network properties [1] that cannot be handled in entirety. Though sampling population is a statistically established area, it is not much explored in the current scenario of real-time dynamic networked data. A comprehensive survey on sampling network streams can be found in [2]. However it does not focus on multi or weighted graphs. We presented in the section III some dynamic sampling mechanisms.

Another problem discussed here is Dynamic Change Detection. Given a temporal network discussed above, how can we detect structural changes across different time steps with an online approach, under the one-pass constraint of data? According to Gama et al. [3], change detection refers to techniques and mechanisms for explicit drift detection characterized by the identification of change points or small time intervals during which changes occur. In evolving networks context, these changes can be detected observing the whole network, for instance communities [4] and motifs [5] evolution; or the changes can be analyzed in a node-centric way, where nodes centrality and roles are observed during network evolution [6]. In section IV, we discussed these techniques.

II. Literature Review

There are a number of research works on sampling static networks but here we will only discuss dynamic sampling algorithms on temporal/evolving network streams coherent to the scope of this article. As most of the real world networks follow power-law distributions in their degree, clustering coefficient etc. From this kind of networks that posses low number of nodes with high degree and high number of nodes with low degree, we are likely to get samples with most or all of nodes from this lighter long tail. Since traditional stratified sampling does not hold with the constraint of one pass, fast and space efficiency, we need to find some strategies that overcome these biases in an efficient way.

In [2] Ahmed et al. presented a simple edge stream sampling which uses a similar approach as reservoir sampling [7] (refer section III-B2). In this case, a new edge enters into the reservoir if its hash value is within top-m minimum hash values, where m is the size of reservoir. However, the method did not ensue as an efficient representative sample. Additionally, Ahmed et al. [2] also proposed a *Partially-induced edge sampling algorithm* called (PIES). This algorithm works by storing nodes and also edges probabilistically in their reservoirs while deleting the one already present at random as in the reservoir sampling [7]. It maintains a fixed size reservoir of nodes while the reservoir size for edges varied based on nodes. CPIES, an update over PIES is given by Zhang et al. [8]. They modified the decremental module of PIES, by deleting the nodes from the reservoir with minimal degree to produce a better cluster preserving structure. PIES had a selection bias to high degree nodes which enhances in CPIES as it tends to delete low degree nodes. Papagelis et al. [9] proposed sampling algorithms that given a user in a social network quickly obtains a near-uniform random sample of nodes in its neighborhood using random walks.

A. Change Detection in Dynamic Networks

Processing graphs as streams is an incoming problem. The work [5] is one of the most complete when considering data mining in evolving graph streams. The focus is on mining

Shazia Tabassum and João Gama are with INESC TEC, University of Porto, Portugal; and Fabíola S. F. Pereira is with Federal University of Uberlandia, Brazil. Corresponding Author e-mail: (jgama@fep.up.pt).

closed graphs, not on change detection though. In [10] a framework for processing graphs as streams is proposed for the link prediction task. This framework considers the cumulative grown of the graph, not addressing the space saving issue [11].

The most studied events in dynamic networks are anomalies and bursts [4]. Anomaly detection refers to the discovery of rare occurrences in data sets [12]. The most representative work in anomaly detection for dynamic graphs is [13]. It addresses the problem considering a time sequence of graphs (graph sequences). The focus is on faults occurring in the application layer of web-based systems. First, they extract activity vectors from the principal eigen vector of dependency matrix. Next, via singular value decomposition, it is possible to find a typical activity pattern (in $t - 1$) and the current activity vector (t). In the end, the angular variable between the vectors defines the anomaly metric. The network processing is through snapshots, not in a streaming fashion. Akoglu and Faloutsos [14] used the Eigen Behavior based Event Detection (EBED) method to detect events in SMS interactions a *who-texts-whom network*. They are able to detect events in a global perspective of the network.

Eberle et al. [15] proposed to discover anomalous sub graphs in graph streams using a change detection metric. The authors compute graph properties GP as the graph evolves and then compare, using average and standard deviation, if there is an abrupt change in these GP. If yes, the change has been detected. The algorithm processes incoming edges in batches using sliding window strategy.

III. Sampling Evolving Multi-Graphs

Evolving Network Stream: In a streaming scenario, a temporally evolving network is usually considered as a stream of edges$\{e_1, e_2, e_3, e_4 ...\} \in E$ generating from a graph stream G. Every edge $e = (u, v, t)$ is composed of a pair of vertices's from V and a time-stamp t, which indicates the time of occurrence of e. We assume that the edges are streaming in the order of time-stamps. E and V can have a temporally changing cardinality.

Multi-graph Stream In a multi-graph stream an edge e can recur randomly in G at various time-stamps t. Examples include phone calls, tweets, co-authorship etc.

Graph Size Graph size in evolving networks is usually the number of edges $|E|$ at any time t or a time interval τ.

A. Methods of Sampling Streaming Graphs

1) Node Based Methods: Node based methods in general, sample a set of nodes from the original graph. The resultant samples contain a set of vertices from the graph stream and showing no connections between them. Acquiring the corresponding edges between them increases the time complexity. Some sampling methods ([16],[8]) store the set of nodes and also the set of edges (which also contain nodes) for ease of computation.

2) Edge Based Methods: These samples are generated by selecting a subset of edges from the original graph. The resultant graph is a subgraph of original graph with nodes and edges. Edges can be labelled, weighted or attributed.

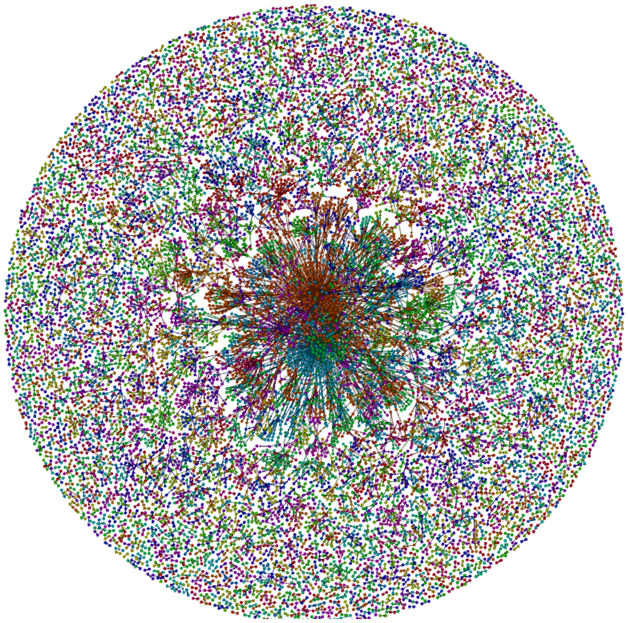


Fig. 1: Pictorial representation of 10^4 top K edges sample at the end of 31 days stream of telecom phone calls (edges) using Space Saving (colors represent communities).

B. Algorithms for Sampling

1) Space Saving Algorithm (SS): What if we need a sample of most active connections in the network at every time stamp t, without having enough space to store all the connections in the network? In such cases the Space Saving Algorithm proposed by [17] is an appropriate choice. Space saving algorithm is the most approximate and efficient algorithm for finding top frequent elements from a data stream. The algorithm maintains partial interest of information as it monitors only a subset of elements from the stream. Considering the edge stream [18], [19] it maintains counters for every element in the sample and increments its count when the edge re-occurs in the stream. If a new edge is encountered in the stream it is replaced with an edge of the least counter value and its count is incremented. Consequently it gives the top K frequent edges at any t from a multi-graph stream. Note that this algorithm keeps track of the top frequent edges but not how much frequent they are. The samples posses a good community structure [18]. Top K, i.e the sample size in this case should be given manually, which do not grow with the growth of the network. Figure 1 shows a sample from a stream of anonimised phone calls provided by a service provider with around 280 to 10 calls per second at mid-day and mid-night and on an average 12M calls per day made by 4M subscribers.

2) Reservoir Sampling (RS): This is a well known algorithm of Reservoir Sampling [7]. This algorithm works by maintaining a reservoir of edges with a predefined sample of size K. In the edge streaming scenario, firstly the reservoir is filled with the initial edges from the stream. Every edge coming after that is computed for the probability K/i of being inserted. Where i is the length of the stream exhausted till then. If the probability of the contending edge in the stream is greater than the probability of an edge in the reservoir which is $1/i$, then uniformly at random an edge is picked from the reservoir. The picked edge is replaced with the edge in the stream. In case the probability is less, the streaming edge is discarded. As i increases, the probability of i^{th} element getting inserted into the reservoir decreases. Therefore, it leads to samples with very old elements from the stream. [18] and [20] show that these samples posses very weak community structure and high bias to low degree nodes.

3) Biased Random Sampling (BRS): This algorithm is proposed as a simple variant of RS in [18]. It is actually an unbiased random sampling technique but considering RS as a standard, this is its biased version. Unlike the above algorithm where the probability of streaming edges diminishes as the stream progresses, this algorithm ensures every edge goes into the reservoir. An edge from the reservoir is chosen for replacement at random. Therefore, the edge insertion is deterministic but deletion is probabilistic. An edge staying for a long time in the reservoir has the same probability of getting out as an edge inserted recently. Consequently, the edges in the reservoir are distributed randomly over time. It gives a better community structure and distributions close to true network than Reservoir Sampling [18] and [20].

The complexity of algorithm increases linearly with the size K of sample in the above mechanisms.

4) Biased Dynamic Sampling Using Forgetting (SBias): Recent interactions are evident to show the current status of relationships, nevertheless some old stronger relations are also substantially significant [20]. Therefore, this sampling algorithm uses a fast memory-less forgetting function with two parameters that help introduce biases on the network based on time and relationship strengths. The main idea is to exponentially forget edges based on time and weight of edges. At every time interval τ the frequency of an edge in a multi-graph is mapped to its weight. Every edge is considered a vector stream of its occurrence and non-occurrence at every τ. The forgetting function is imposed on all $|E|$ edge vector streams independently, where E is an edge set at time interval τ. An illustration is given in figure 3. A threshold θ is used to eliminate the edges from the network. The parameter values α and θ can be increased to decrease the sample size.

$$\hat{w}_\tau(e) = w_\tau(e) + (1-\alpha)\hat{w}_{\tau-1}(e) \tag{1}$$

Where $w_\tau(e)$ is the weight of an edge at τ. This algorithm provides better distributions (closest to the true network) than the above algorithms (figure 8) [20]. This illustration is given using the CollegeMsg Networked data set comprised of private messages sent on an online social network at the University of California, Irvine [21], obtained from SNAP data sets[2]. One of the important property of this algorithm is it does not maintain a fixed size sample, which gets decreasing with the increasing size of true network (most of them obeying a power law increase over time). Though the sample size varies it is bounded by the variance of network size per time step.

C. Sampling Ego-Networks with forgetting factor

As ego-networks also densify [22] over time by making infeasible to store all the information of them in the memory. An example graph of an ego-network (2-levels) from a phone calls network is shown in figure 2. The function (1) and forgetting mechanism here is the same as in the above SBias model but applies for a single ego/personal network with any number of levels/radius from the ego. The main variation is that in the above algorithm we don't need to remove the edges adjacent to the deleted edge. In this case we remove the edges adjacent to a deleted edge which otherwise do not have a connection to the ego [23]. This method has an advantage as an input for detecting changes in personal networks, predicting links and detecting frauds etc.

IV. Change Detection in Evolving Networks

We present here the change detection techniques from a node-centric perspective in a network stream processing environment. We call this task as *Node Centrality Change Detection*. The techniques here presented were proposed in [6] and focus on tracking nodes properties instead of global graph properties ([15], [14]).

In the approaches presented in this section, we consider an edge stream S which is a continuous and unbounded flow of objects $E_1, E_2, E_3.....$, where each edge E_i is defined by (v, u, t) which represents a connection between vertices/nodes u and v at time t. The vertices $\{u, v, ...\} \in V$ and get added to or deleted from V at anytime t.

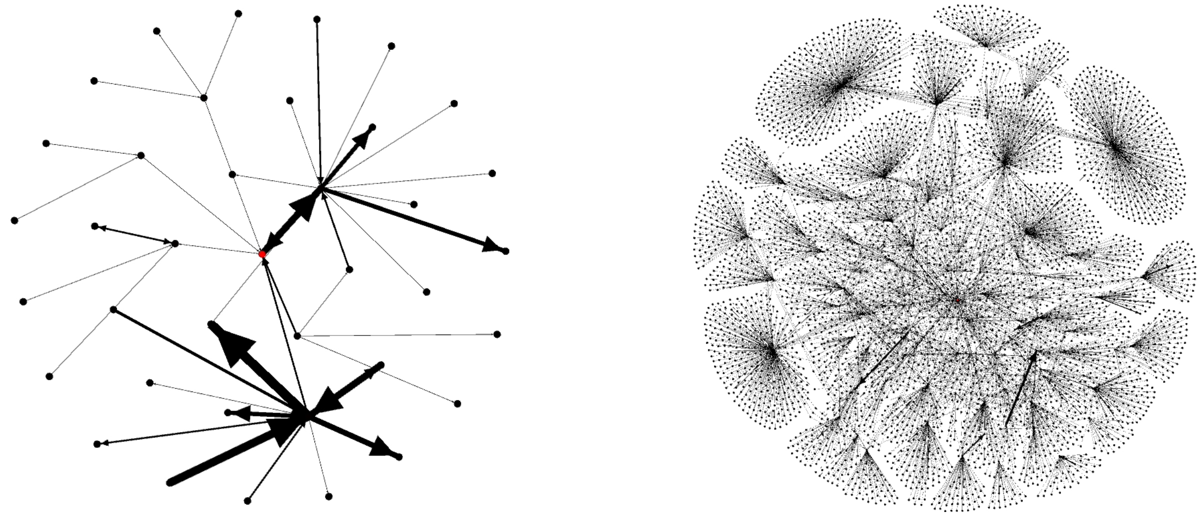
A. Processing a Streaming Network

For every incoming edge (v, u, t) from a stream S, the centrality scores $C^m(v)$ and $C^m(u)$ for nodes u and v are updated in the order of t, for m being a node centrality metric (in case of degree centrality, the degree of nodes u and v is updated for every incoming edge $\{u, v\}$ at t where as in the case of betweenness and closeness, the centrality of nodes are updated only after every T). Another variable T is a discrete time-step/time-interval with granularity defined by the user. After every T the centrality scores are reset and starts accumulating again. Therefore, we keep track of centrality score of nodes per day. Consequently we store a set of nodes (with changing cardinality) and a streaming vector of its associated centralities per time step T. As a result, we have an independent non stationary stream of centrality scores $\{C^m_{T_1}, C^m_{T_2}, C^m_{T_3}........\}$ for every node v in S after every time step T. To get a normalized version of scores, after every time-step T the centrality of a node is divided by the number of nodes in graph at T. Therefore we have normalized centrality scores in the vector stream. Further we employed aggregating mechanisms to the above streams of centrality scores per node.

B. Aggregating Mechanisms

For notational simplicity in the below equations we use C_T for $C^m_T(v)$ as all notations for the techniques below are considered for a stream of centrality scores per node per centrality metric.

[2] Data available at https://snap.stanford.edu/data/CollegeMsg.html



(a) (b)

Fig. 2: Ego-network of a user (red) on day 1 and day31 from the phone call network.

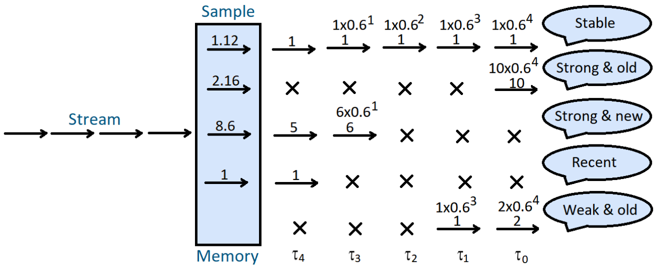


Fig. 3: Pictorial representation of SBias sample at τ_4 with $\alpha = 0.4$ and $\theta = 0.5$.

1) Moving Window Average (MWA): A window of size W_S consists of data points from the latest temporal time steps $\{T, T-1, T-2, ..., T-(W_S-1)\}$. The window keeps on sliding to always maintain the latest W_S time steps and the data points from $T - W_S$ are forgotten. Alongside, the mean of data points within the window is calculated by using simple equation (2) where C_{T-i} is the stream of centrality scores at time-step $T-i$ using measure m per node. In this approach all the data points in the window are assigned equal weights.

$$\mu_T = \frac{1}{W_S} \sum_{i=0}^{W_S-1} C_{T-i} \tag{2}$$

As the window slides the mean of data points in the window is updated, either using the above equation (2) for small window sizes and equation (3) for large window sizes.

$$\mu_T = \mu_{T-1} W_S - C_{T-W_S} + C_T \tag{3}$$

2) Weighted Moving Window Average (WMWA): Weighted moving window average follows the same window sliding strategy as in MWA and computes average over the data points in the window. The improvement over MWA is that the accumulated data points per time step T in the window are weighted linearly as given in equation (4). The oldest data points in the window attain a least weight and the latest data point acquires the highest weight linear to the least one. Weights are updated, when the window slides. Assignment of weights per data point depends on the size of window.

$$\mu_T = \sum_{i=0}^{W_S-1} \frac{C_{T-i}(W_S - i)}{W_S - i} \tag{4}$$

3) Page Hinckley Test (PH): Page Hinckley [24] is one of the memory less sequential analysis techniques typically used for change detection [25], [26], [27], [3]. We use it as a non-parametric test, as the distribution is non stationary and not known. This test considers a cumulative variable m_T, defined as the cummulated difference between the latest centrality score at T and the previous mean till the current moment, as given in the equation (5) below:

$$m_T = \sum_{i=1}^{T} |C_T - \mu_{T-1}| - \alpha \tag{5}$$

Where $\mu_T = 1/|T| \sum_{i=1}^{T} C_i$, $\mu_0 = 0$ and α = magnitude of changes that are allowed. For calculating μ_T we also need to store the number of time-steps passed.

The equation (5) given above uses fixed α value, which is not pertinent with our multiple vector streams of centralities per node, where the centrality scores of few active nodes are way higher than some least active nodes. Therefore, using same value of α over differing node centralities would not be fair enough. Hence, we use a relative α, which is relative with the differing centrality scores per node. Relative α is a point percentage of previous aggregated mean of that node, as given in equation (6).

$$m_T = \sum_{i=1}^{T} |C_T - \mu_{T-1}| - \alpha\mu_{T-1} \tag{6}$$

Further to calculate change point score we need a variable M_T which is the minimum value of m_T and is always maintained and updated for every new time step T as given in equation 7

$$M_T = min(m_T; i = 1...T) \tag{7}$$

C. Detecting Change Points

1) Change Point Scoring Function: To detect the change points and their magnitude after every time-step T in MWA and WMWA, we use a change point scoring function given in equation (8)

$$\Gamma_T = \frac{|C_T - \mu_{T-1}|}{max(C_T, \mu_{T-1})} \tag{8}$$

Where C_T is the current centrality score and μ_{T-1} is the mean of previous centrality scores in the window. The change point scoring function gives the percentage point increase or decrease of the current centrality score with the previous mean. It takes values $0 \leq \Gamma_T \leq 1$.

For a PH test, after every time-step T the change points are scored using the equation (9).

$$\Gamma_T = m_T - M_T \tag{9}$$

2) Change Point Detection: We can decide the magnitude of change allowed by the above change point scoring function. For this we use a threshold θ on Γ, to signal an alarm of change in the node. It takes values either 0 or 1. ”1” indicates a node centrality change and ”0” indicates no change.

$$\epsilon_T = \begin{cases} 1, & \text{if } \Gamma_T \geq \theta. \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

We also apply a relative θ for detecting change points in PH Test only, as the change point scores from windowed approaches are already normalized in equation (8). Therefore to normalize threshold over multiple streams of centrality scores in PH test we use a relative threshold θ by multiplying the threshold θ with M_T of that node at time T as in equation 11.

$$\epsilon_T = \begin{cases} 1, & \text{if } \Gamma_T \geq (\theta \times M_T). \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

While carrying out the above-mentioned mechanisms we considered the following assumptions. For window based approaches change detection starts only after the window of size W_S is filled. If there exists no edges for a node in a time step T, then the mean is calculated assuming a ”0” centrality score. If a node is newly introduced (with edges) in the stream in the time interval T then the previous mean at $T-1$ is considered ”0” during change point scoring. In window based approaches if a node does not appear in the stream for a W_S time steps, the node is deleted to save space.

The above described model for change detection using centrality measures is illustrated in figure 4

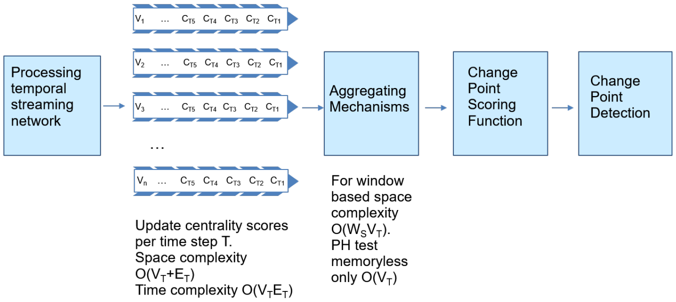

V_1 ... C_{T5} C_{T4} C_{T3} C_{T2} C_{T1}
V_2 ... C_{T5} C_{T4} C_{T3} C_{T2} C_{T1}
V_3 ... C_{T5} C_{T4} C_{T3} C_{T2} C_{T1}
...
V_n ... C_{T5} C_{T4} C_{T3} C_{T2} C_{T1}
Processing temporal streaming network
Aggregating Mechanisms
Change Point Scoring Function
Change Point Detection
Update centrality scores per time step T. Space complexity $O(V_T+E_T)$ Time complexity $O(V_TE_T)$
For window based space complexity $O(W_SV_T)$. PH test memoryless only $O(V_T)$

Fig. 4: Preference change detection model.

D. Centrality changes in Twitter network

In [28], [6] we applied the node centrality change detection problem to infer user preference changes in temporal interaction networks. Figure 5 illustrates the Twitter evolving network related to Brazilian news we utilized. The network represents interactions among users through retweets.

In order to exemplify the techniques presented in this section, Figure 6 shows the events detected in Twitter network from Figure 5 based on in-degree centrality and MWA aggregating mechanism.

V. Temporal Centralities in Temporal Social Networks

The topological structure of static networks can be characterized by an abundance of measures. In essence, such measures are based on connections between neighboring nodes (such as the degree or clustering coefficient), or between larger sets of nodes (such as path lengths, network diameter and centrality measures). When the additional dimension of time is included in the network picture, many of these measures need rethinking. Our main goal here is to represent social networks, specially Twitter, founded on temporal graphs theory [29]. According to Holme and Saramaki [30], temporal networks can be divided into two classes corresponding to the types of representations: contact sequences and interval graphs. While in contact sequences, the edges are active over a set of times, in interval graphs they are active over a set of intervals. In Figure 7 we exemplify Twitter social network as an interval graph.

In temporal networks the concept of geodesic distance should take into account the temporal ordering of links [30]. A temporal path $P_{u,v}$ in a temporal graph G is a sequence $P_{u,v} =< (v_1, v_2, t_1), (v_2, v_3, t_2), ..., (v_{k-1}, v_k, t_{k-1}) >$, where $(v_i, v_{i+1}, t_{init}, t_{end}) \in E$ is the i-th temporal edge on $P_{u,v}$, $1 \leq i \leq k$, R is the retention time of nodes, i.e., the time between information arrival in the node and the instant from which it can be forwarded, T is the edge traversal time, $t_i + R + T \leq t_{i+1}$, $t_{init} \leq t_i \leq t_{end}$, $n \leq t_1$ and $t_{k-1} \leq N$, $u = v_1$ and $v = v_k$.

In a Twitter temporal graph representation, one can adopt $R = 1$ day and $T = 0$, as tweets are published instantaneously and the average interaction time for posts is one day [31]. Considering the temporal network of Figure 7 and the parameters $W = [1, 9]$, $T = 0$ and $R = 1$, we can cite some examples

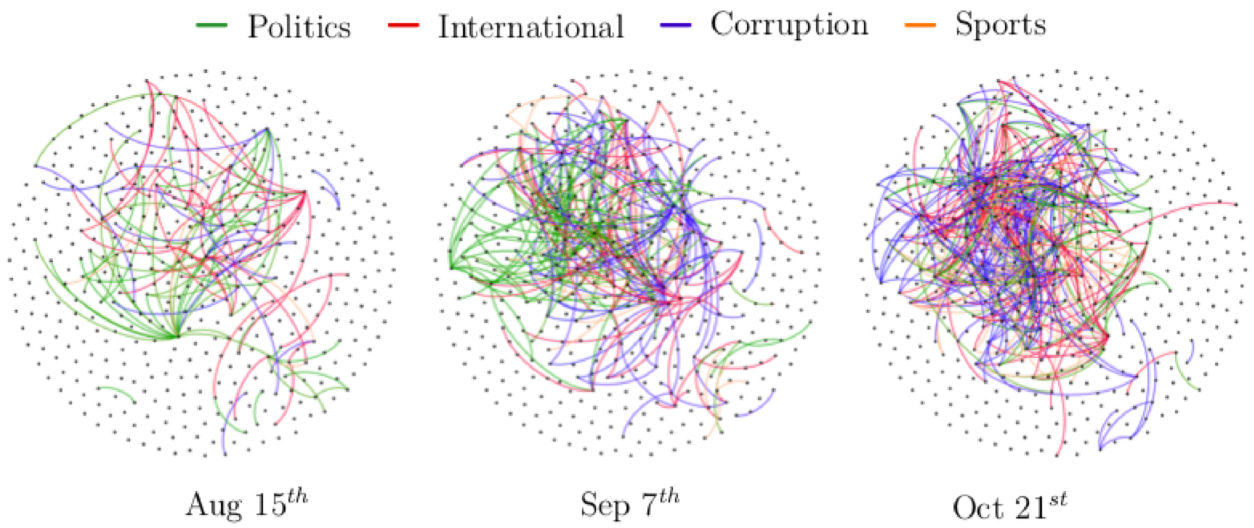


Fig. 5: Snapshots of samples of the evolving interaction network. Nodes are Twitter users. One tie from user u_1 to u_2 means that u_2 retweed at t some text originally posted by u_1. Colors represent topics that users are talking about. The samples were built by filtering nodes with degree between 50-22000 and edges representing the 4 most popular topics. Each snapshot corresponds to 1 day time-interval. This figure highlights the *edges* evolving aspect. Nodes are not evolving for better visualization.

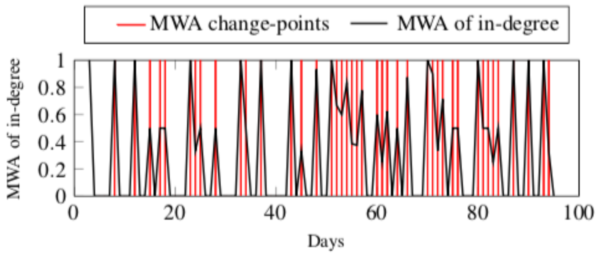


Fig. 6: Change events detected (red) in Twitter network based on in-degree centrality and MWA aggregating mechanism.

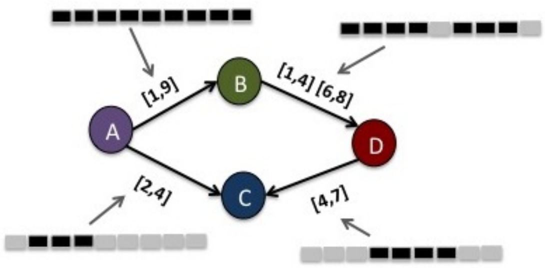


Fig. 7: Twitter as an interval graph. Nodes are Twitter users and an edge $(u, v, t_{init}, t_{end})$ indicates that v starts following u at t_{init} and unfollows u at t_{end+1}.

of temporal paths: $P_{A,D} =< (A, B, 1), (B, D, 2) >= 1$ (fastest path), $P_{A,D} =< (A, B, 2), (B, D, 6) >= 4$, $P_{A,C} =< (A, C, 2) >= 0$ (fastest path).

Centrality metrics that take into account the distance between two nodes are impacted by temporal paths definition. We focus on closeness and betweenness [32]. To compute these metrics in a temporal network scenario, we need to consider the number of fastest paths instead of the number of shortest paths as in static definition. A comprehensive study on centrality metrics for temporal networks can be found on [33].

As application example of temporal centralities in temporal networks, in [28] we discuss that tracking how follow/unfollow relationships on Twitter evolve over time can help us to understand their impact on users behaviors.

VI. Conclusion

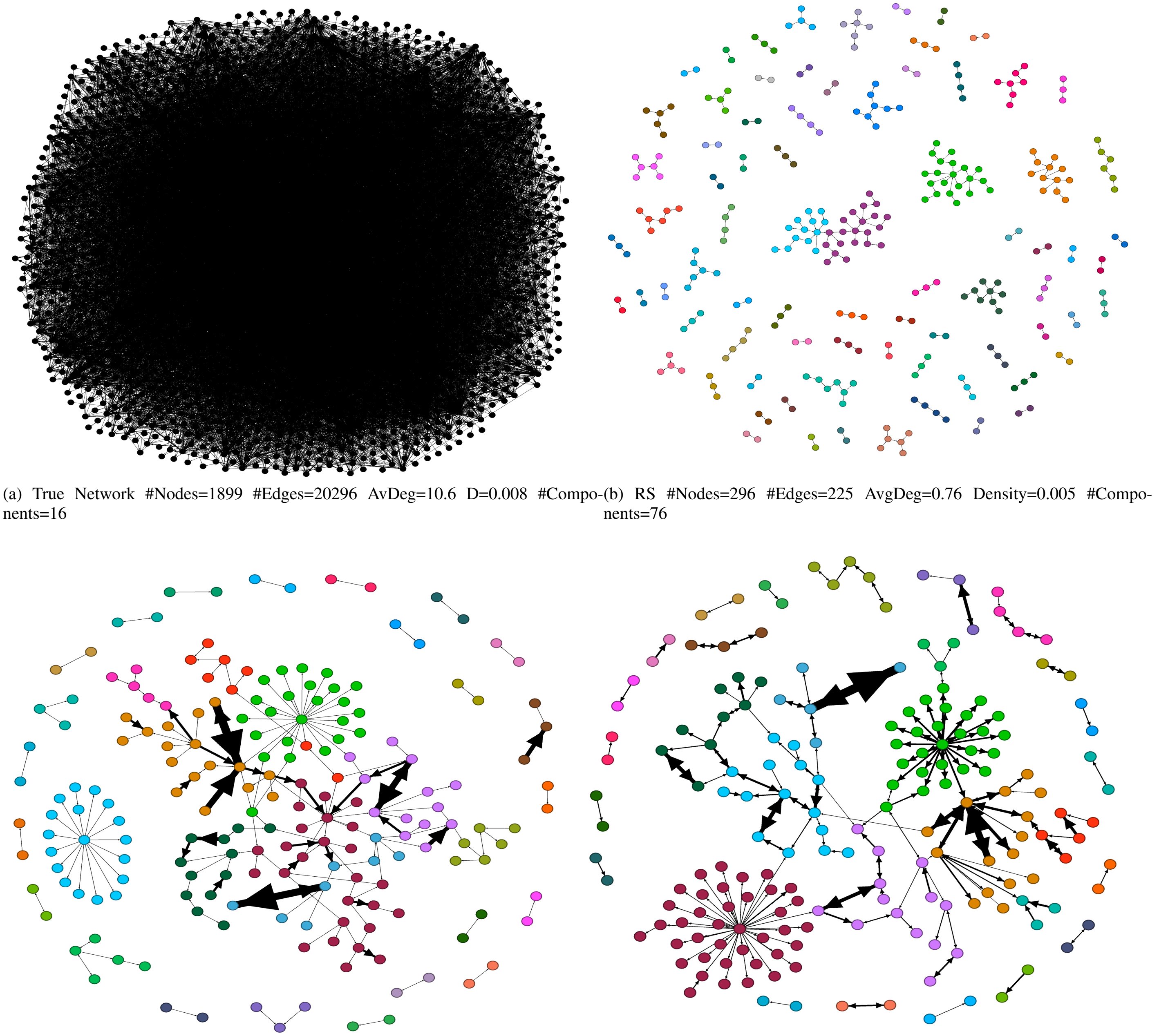
We discussed three topics in this article concerned from the size and complexity of the networks in the start to finish of knowledge discovery process. Four sampling mechanisms for fast massive scale free networks were presented, namely Space Saving for top K, Reservoir Sampling, Biased Random Sampling and Sampling using forgetting (SBias). Their properties and biases were discussed supported with some illustrations for better comprehensibility.

Further we demonstrated a fast approach for detecting changes in the network using centralities alternative to node features or content that is time and space expensive to be acquired. Additionally these are complimented with some aggregating and memory less tests for efficient change points detection.

Lastly some concepts relating to information transmission and temporal paths were discussed briefly, opening new range of possibilities in these kind of networks.

Acknowledgment

The authors would like to thank the support of the project TEC4Growth RL SMILES Smart, mobile, Intelligent and Large Scale Sensing and analytics NORTE-01-0145-FEDER-000020 which is financed by the North Portugal regional operational program (NORTE 2020), under the Portugal 2020 partnership agreement, and through the European regional development fund.



(a) True Network #Nodes=1899 #Edges=20296 AvDeg=10.6 D=0.008 #Components=16

(b) RS #Nodes=296 #Edges=225 AvgDeg=0.76 Density=0.005 #Components=76

(c) BRS #Nodes=189 #Edges=225 AvgDeg=1.196 Density=0.006 #Components=25

(d) SBias #Nodes=182 #Edges=225 AvgDeg=1.236 Density=0.007 #Components=21

Fig. 8: Snapshot at the end of observed stream (CollegeMsg) of true network and samples (1%).

References

[1] N. K. Ahmed, N. Duffield, T. L. Willke, and R. A. Rossi, “On sampling from massive graph streams,” *Proceedings of the VLDB Endowment*, vol. 10, no. 11, pp. 1430–1441, 2017.

[2] N. K. Ahmed, J. Neville, and R. Kompella, “Network sampling: From static to streaming graphs,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 2, p. 7, 2014.

[3] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

[4] M. Cordeiro and J. Gama, *Online Social Networks Event Detection: A Survey*. Cham: Springer International Publishing, 2016, pp. 1–41.

[5] A. Bifet, G. Holmes, B. Pfahringer, and R. Gavaldà, “Mining frequent closed graphs on evolving data streams,” in *17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’11, 2011, pp. 591–599.

[6] F. S. F. Pereira, S. Tabassum, J. Gama, S. de Amo, and G. M. B. Oliveira, *Processing Evolving Social Networks for Change Detection Based on Centrality Measures*. Cham: Springer International Publishing, 2019, pp. 155–176.

[7] J. S. Vitter, “Random sampling with a reservoir,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 11, no. 1, pp. 37–57, 1985.

[8] J. Zhang, K. Zhu, Y. Pei, G. Fletcher, and M. Pechenizkiy, “Clustering-structure representative sampling from graph streams,” in *International Workshop on Complex Networks and their Applications*. Springer, 2017, pp. 265–277.

[9] M. Papagelis, G. Das, and N. Koudas, “Sampling online social networks,” *IEEE Transactions on knowledge and data engineering*, vol. 25, no. 3, pp. 662–676, 2013.

[10] J. Fairbanks, D. Ediger, R. McColl, D. A. Bader, and E. Gilbert, “A statistical framework for streaming graph analysis,” in *IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, ser. ASONAM ’13, 2013, pp. 341–347.

[11] J. Gama, *Knowledge discovery from data streams*. CRC Press, 2010.

[12] L. Akoglu, H. Tong, and D. Koutra, “Graph based anomaly detection and description: a survey,” *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[13] T. Ide and H. Kashima, “Eigenspace-based anomaly detection in computer systems,” in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’04, 2004, pp. 440–449.

[14] L. Akoglu and C. Faloutsos, “Event detection in time series of mobile communication graphs,” in *Proceedings of 27th army science conference*, ser. 18, vol. 2, no. 3, 2010.

[15] W. Eberle and L. Holder, “Identifying anomalies in graph streams using change detection,” in *KDD Workshop on Mining and Learning in Graphs (MLG)*, 2016.

[16] N. K. Ahmed, J. Neville, and R. Kompella, “Space-efficient sampling from social activity streams,” in *Proceedings of the 1st international workshop on big data, streams and heterogeneous source mining: algorithms, Systems, Programming Models and Applications*. ACM, 2012, pp. 53–60.

[17] A. Metwally, D. Agrawal, and A. El Abbadi, “Efficient computation of frequent and top-k elements in data streams,” in *International Conference on Database Theory*. Springer, 2005, pp. 398–412.

[18] S. Tabassum and J. Gama, “Sampling massive streaming call graphs,” in *ACM Symposium on Advanced Computing*, 2016, pp. 923–928.

[19] S. Tabassum, “Social network analysis of mobile streaming networks,” in *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, vol. 2. IEEE, 2016, pp. 20–25.

[20] S. Tabassum and J. Gama, “Biased dynamic sampling for temporal network streams,” in *Complex Networks and Their Applications VII. COMPLEX NETWORKS 2018. Studies in Computational Intelligence, vol 812*. Springer, 2018, pp. 512–523.

[21] P. Panzarasa, T. Opsahl, and K. M. Carley, “Patterns and dynamics of users’ behavior and interaction: Network analysis of an online community,” *Journal of the American Society for Information Science and Technology*, vol. 60, no. 5, pp. 911–932, 2009.

[22] S. Tabassum and J. Gama, “Evolution analysis of call ego-networks,” in *International Conference on Discovery Science*. Springer, 2016, pp. 213–225.

[23] ——, “Sampling evolving ego-networks with forgetting factor,” in *Workshop MobDM, Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, 2016.

[24] E. S. Page, “Continuous inspection schemes,” *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[25] H. Mouss, D. Mouss, N. Mouss, and L. Sefouhi, “Test of page-hinkley, an approach for fault detection in an agro-alimentary production system,” in *Proceedings of the Asian control conference*, vol. 2. Citeseer, 2004, pp. 815–818.

[26] J. Gama, R. Sebastião, and P. P. Rodrigues, “On evaluating stream learning algorithms,” *Machine learning*, vol. 90, no. 3, pp. 317–346, 2013.

[27] R. Sebastião, M. M. Silva, R. Rabiço, J. Gama, and T. Mendonça, “Real-time algorithm for changes detection in depth of anesthesia signals,” *Evolving Systems*, vol. 4, no. 1, pp. 3–12, 2013.

[28] F. S. Pereira, J. a. Gama, S. Amo, and G. M. Oliveira, “On analyzing user preference dynamics with temporal social networks,” *Mach. Learn.*, vol. 107, no. 11, pp. 1745–1773, Nov. 2018.

[29] F. S. F. Pereira, S. Amo, and J. Gama, “Evolving centralities in temporal graphs: a twitter network analysis,” in *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, 2016.

[30] P. Holme and J. Saramaki, “Temporal networks,” *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012.

[31] H. Wu, J. Cheng, S. Huang, Y. Ke, Y. Lu, and Y. Xu, “Path problems in temporal graphs,” *Proceedings of the VLDB Endowment*, vol. 7, no. 9, pp. 721–732, 2014.

[32] R. Zafarani, M. A. Abbasi, and H. Liu, *Social Media Mining: An Introduction*. New York, NY, USA: Cambridge University Press, 2014.

[33] V. Nicosia, J. Tang, C. Mascolo, M. Musolesi, G. Russo, and V. Latora, *Temporal Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ch. Graph Metrics for Temporal Networks, pp. 15–40.

Network Science of Teams: Current State and Future Trends

Liangyue Li and Hanghang Tong

***Abstract*—Teams are increasingly indispensable to achievements in any organization. Despite the organizations substantial dependency on teams, fundamental knowledge about the conduct of team-enabled operations is lacking, especially at the social, cognitive and information level in relation to team performance and network dynamics. Generally speaking, the team performance can be viewed as the composite of its users, the tasks that the team performs and the networks that the team is embedded in or operates on. The goal of this article is to (1) provide a comprehensive review of the recent advances in optimizing teams performance in the context of networks; and (2) identify the open challenges and future trends. We believe this is an emerging and high-impact topic in computational social science, which will attract both researchers and practitioners in the data mining as well as social science research communities. Our emphasis will be on (1) the recent emerging techniques on addressing team performance optimization problem; and (2) the open challenges/future trends, with a careful balance between the theories, algorithms and applications.**

***Index Terms*—Network science of teams, team performance characterization, performance prediction, team optimization.**

I. Introduction

IN defining the essence of professional teamwork, Hackman and Katz [1] stated that teams function as 'purposive social systems', defined as people who are readily identifiable to each other by role and position and who work interdependently to accomplish one or more collective objectives. Teams are increasingly indispensable to achievements in any organization. This is perhaps most evident in multinational organizations where communication technology has transformed the geographically dispersed teams and networks. Business operations in the large organizations now involve large, interactive, and layered networks of teams and personnel communicating across hierarchies and countries during the execution of complex and multifaceted international businesses. Despite the organizations' substantial dependency on teams, fundamental knowledge about the conduct of team-enabled operations is lacking, especially at the *social, cognitive* and *information* level in relation to team performance and network dynamics. What do high-performing engineering/design/sale teams share in common with respect to their communication patterns? How to predict a team's performance before it starts to work on the assigned project? How to foster productive behavioral changes of team members and leaders in order to optimize performance?

Generally speaking, the **team performance** can be viewed as the composite of the following three aspects, including (1) its users, (2) tasks that the team performs and (3) the networks that the team is embedded in or operates on. In this article, we will provide a comprehensive review of the recent advances in *characterizing*, *predicting* and *optimizing* teams' performance in the context of composite networks (i.e., social-cognitive-information networks).

Research in sociology and psychology has long been trying to characterize the high-performing teams in organizations. The basics of team effectiveness were identified by J. Richard Hackman, who uncovered a groundbreaking insight: what matter most to collaboration are certain enabling conditions. Recent studies find that three of Hackman's conditions – a compelling direction, a strong structure, and a supportive context – continue to be particularly critical to team success [2]. We would comprehensively survey related literatures in sociology, psychology and computer science.

Understanding the dynamic mechanisms that drive the success of high-performing teams can provide the key insights into building the best teams and hence lifting the productivity and profitability of the organizations. For this purpose, we introduce some of the recent work on developing novel predictive models to forecast the long-term performance of teams (*point prediction*) as well as the pathway to impact (*trajectory prediction*). It is also worthwhile to quantitatively examine the relationship between the team level and individual level performances to build a joint predictive model.

From the practical perspective, it is important to form a good team in the context of networks for a given tasks. For an existing team, it is often desirable to optimize its performance through expanding the team by bringing a new team member with certain expertise, finding a new candidate to replace a current under-performing team member or downsizing the team for the purpose of cost reduction. We would introduce recent advances in team performance optimization.

II. Team Performance Characterization

A. Collective Intelligence

The notion of individual intelligence was first proposed by Charles Spearman when he noticed that school kids who did well in one school subject tend to do well in many other school subjects [3]. The observations that the average correlation among individual's performance on a variety of cognitive tasks is positive and the first factor extracted using a factor analysis accounts for about 30-50% of the variance indicate the existence of general intelligence. The first factor is

The authors are from the School of Computing, Informatics, Decision Systems Engineering, Arizona State University, Tempe, AZ 85281, USA. Corresponding Author e-mail: ({liangyue, htong6}@asu.edu).

usually referred to as general intelligence. We can give people a relatively limited set of items and the scores of these items can predict how they perform across a variety of domains and over a long period of time. Such intelligence test can predict not only how kids do in school in multiple subjects, but also the probability that they would be successful in their future career. This is perhaps the most empirically replicated facts in most of the psychology.

A group of researchers at CMU set out to test whether a similar notion of collective intelligence exists in a team of people, i.e., whether a single factor exists from the team's performance on a variety of tasks [4]. They enlisted 40 and 152 teams of size two to five for their two studies. They assigned a diverse set of group tasks to these teams. The tasks can be categorized into four types, namely, 'generate', 'choose', 'negotiate', and 'execute'. The results support their initial hypothesis that the average correlation among the teams' scores on the diverse set of tasks is positive and the factor analysis reveals that one single factor can account for more than 43% of the variance. Additionally, the collective intelligence score calculated using the first factor can strongly predict the team's performance on a future criterion tasks (e.g., video game and architectural design). Surprisingly, the average team member intelligence and the maximum team member intelligence are not that predictive of the future performance, which tells us that simply assigning a team of smart people does not promise a smart team. But what are the ingredients that are important to an intelligent team? Surprisingly, the team processes, e.g., group cohesion, motivation, and satisfacation, traditionally regarded as important to team performance, are not predictive of collective intelligence. The collective intelligence is found to be positively correlated with the average social perceptiveness of the team members and negatively correlated with the variance in the number of speaking turns by team embers.

B. Virtual Teams in Online Games

The above research about collective intelligence (CI) is mainly on traditional teams where team members have face to face interactions. It would be interesting to examine whether the collective intelligence also exists in virtual teams. Virtual teams are diverse, dispersed, digital and dynamic, e.g. the Multiplayer Online Battle Arena (MOBA) teams. Considering that such teams perform tasks at a fast pace without explicit face-to-face or verbal communitication, other means of coordinations might player a more critical role here, e.g., tacit coordiation, or coordinations that happen without explicit verbal communication [5]. Studying how collective intelligence works in such MOBA teams could also inform the operations of other virtual teams commonly seen in business world, where teams are dipsersed across geographical boundaries and making decisions at a fast pace.

One recent study [5] examines collective intelligence in *League of Legends (League)* teams, a popular game with worldwide monthly active user base of 67 million. In *League*, a match is between two teams of five members and teams can be formed either through the game's matchmaking algorithms or by recruiting other players in the game community. One team's goal is to destroy the opponent team's base. The authors hypothesize that (1) CI will predict team performance in *League*, (2) social perceptiveness and proportion of woman will be positively associated with CI in *League*, and (3) CI will not be associated with equality of contribution to conversation or decision making in *League* teams. In order to know the CI, game performance, and team characteristics, the authors collect data from three sources: (1) all team members completed a questionnaire on their own about information on their demographics, psychological variables, cognition, affect, etc; (2) the teams took the Test of Collective Intelligence (TCI), an online test battery, as a group to measure the collective intelligence of each team; and (3) the play statistics including the team performances are provided by Riot Games. There were 248 teams that completed all components of the study and 85% of the teams are all males. The authors find that CI also exists in *League* teams from factor analysis and it is positively correlated with the performance measure of the teams controlling for individual and team play time. Besides, CI is positively correlated with the number of woman in the team and is positively correlated with social perceptiveness, but the proportion of woman and social perceptiveness are not correlated. What's interesting is that the equality of communication measured by standard deviation of chat lines and chat word count is not significantly correlated with CI. In addition, CI is negatively correlated with some group process, e.g., perceived equality in decision making, frequency of game-specific communication. These suggest that highly dispersed and dynamic virtual teams tend to adopt a tacit coordination method.

C. Networks in Sports Teams

Recently, a number of works start to examine the network structure in sports teams in relation to their performances [6], [7]. Using Euro Cup 2008 tournament data, researchers construct a directed network of "ball flow" among players in the team [6], where nodes represent players and edge weights indicate the number of successful passes between two players. They use the betweenness centrality of the player with regard to the opponent's goal as the performance measure of a player and the team level performance is defined as the average performance of the top-k players. They find that the difference between two teams' defined performance measure is indicative of their winning probability. In a similar study, researchers use English Premier League soccer team data to find that increased network density among team members lead to increased team performance and increased centralization of team play decreases the performance [7].

D. Networks in GitHub Teams

Social coding platforms such as GitHub offer a unique experience to developers as they can subscribe to activities of other developers. Using GitHub data, researchers construct two types of networks [8]: a project-project network, where nodes represent projects and two nodes are connected if they share at least one common developer; and developer-developer network, were nodes represent developers and two nodes are connected if they have collaborated in the same project. They find that in the project-project network, the diameter of the largest connected component is 9 with the average shortest path 3.7, which is more interconnected than human networks; and in the developer-developer network, the average shortest path is 2.47. Compared with the average shortest path of Facebook 4.7, we see social coding enables substantially more collaborations among developers.

III. Team Performance Prediction

A. Long-term Performance Prediction

For the discussion in this section, we mainly use research teams since their performance can be measured by the impact of their team products (e.g., research papers, patents). Understanding the dynamic mechanisms that drive those high-impact scientific work is a long-debated research topic and has many important implications, ranging from personal career development and recruitment search, to the jurisdiction of research resources. Scholars, especially junior scholars, who could master the key to producing high-impact work would attract more attentions as well as research resources; and thus put themselves in a better position in their career developments. High-impact work remains as one of the most important criteria for various organization (e.g. companies, universities and governments) to identify the best talents, especially at their early stages. It is highly desirable for researchers to judiciously search the right literature that can best benefit their research.

Recent advances in characterizing and modeling scientific success have made it possible to forecast the long-term impact of scientific work. Wuchty et al. [9] observe that papers with multiple authors receive more citations than solo-authored ones. Uzzi et al. [10] find that the highest-impact science work is primarily grounded in atypical combinations of prior ideas while embedding them in conventional knowledge frames. Recently, Wang et al. [11] develop a mechanistic model for the citation dynamics of individual papers. In particular, they identify three fundamental drives underlying the citation histories of individual papers, namely, preferential attachment, temporal citation trend, and fitness. They combine these three factors into a mechanistic model, which fits well on the Physical Review corpus and is able to predict future citations with good accuracy. In data mining community, efforts have also been made to predict the long-term success. Carlos et al. [12] estimate the number of citations of a paper based on the information of past articles written by the same author(s). Yan et al. [13] design effective content (e.g., topic diversity) and contextual (e.g., author's *h*-index, venue's centrality) features for the prediction of future citation counts.

To collectively address a number of key algorithmic challenges, namely, scholarly feature design (C1), non-linearity (C2), domain heterogeneity (C3), and dynamics (C4), in relation to predicting long-term scientific impact, a joint predictive model *iBall* is proposed [14]. First (for C1), they find that the citation history of a scholarly entity (e.g., paper, researcher, venue) in the first three years (e.g., since its publication date) is a strong indicator of its long-term impact (e.g., the accumulated citation count in ten years); and adding additional contextual or content features brings little marginal benefits in terms of prediction performance. This not only largely simplifies the feature design, but also enables them to forecast the long-term scientific impact at its early stage. Second (for C2), their joint predictive model is flexible, being able to characterize both the linear and non-linear relationship between the features and the impact score. Third (for C3), they propose to jointly learn a predictive model to differentiate distinctive domains, while taking into consideration the commonalities among these similar domains (see an illustration in Figure 1). Fourth (for C4), they further propose a fast on-line update algorithm to adapt our joint predictive model efficiently over time to accommodate newly arrived training examples (e.g., newly published papers).

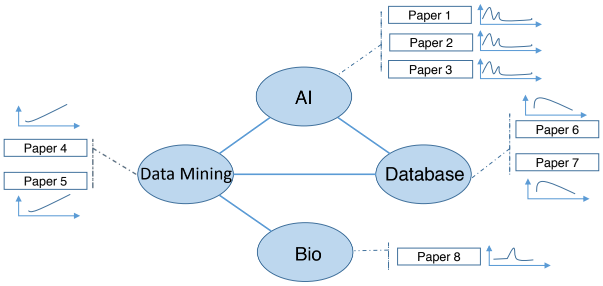


Fig. 1. An illustrative example of the joint predictive model iBall [14]. Papers from the same domain (e.g., AI, Databases, Data Mining and Bio) share similar patterns in terms of attracting citations over time. Certain domains (e.g., AI and Data Mining) are more related with each other than other domains (e.g., AI and Bio). The authors want to jointly learn four predictive models (one for each domain), with the goal of encouraging the predictive models from more related domains (e.g., AI and Data Mining) to be 'similar' with each other.

B. Performance Trajectory Forecasting

From the prediction perspective, more often than not, it is of key importance to forecast the pathway to impact for scholarly entities (e.g., how many citations a research paper will attract in each of several consecutive years in the future). The impact pathway often provides a good indicator of the shift of the research frontier. For instance, the rapid citation count increase of the deep learning papers reveals an emerging surge of this topic. The impact pathway can also help trigger an early intervention should the impact trajectory step down in the near future.

The state of the art has mainly focused on modeling the long-term scientific impact for the early prediction, as we have discussed in the previous subsection. They are not directly applicable to forecasting the impact pathway, e.g., citation counts in each of the next 10 years. One baseline solution is to treat the impacts across different years independently and to train a separate model for each of the impacts. This treatment might ignore the inherent relationship among different impacts across different years, and thus might lead to sub-optimal performance. Having this in mind, a better way could be to apply the existing multi-label/multi-task learning methods to exploit the relation among impacts across different years. Nonetheless, these general-purpose multi-label/multi-task learning approaches might overlook some unique characteristics of the impact pathway prediction.

A new predictive model (*iPath*) is proposed to simultaneously fulfill two design objectives with the unique properties of impact pathway prediction [15]. First, *prediction consistency.* Intuitively, the scholarly impacts at certain years might be correlated with each other, which, if vetted carefully, could boost the prediction performance (i.e., multi-label or multi-task learning). Here, one difficulty for impact pathway prediction is that such a relation structure is often not accurately known a priori. The *iPath* model is capable of simultaneously inferring the impact relation structure from the training data and leveraging such (inferred) relation to improve the prediction performance. Second, *parameter smoothness.* For a given feature of the predictive model, one do not expect its effect on the impacts of adjacent years would change dramatically. The *iPath* model is able to capture such temporal smoothness.

C. Team Performance vs. Individual Performance

The great Greek philosopher Aristotle articulated more than 2,000 years ago that "*the whole is greater than the sum of its parts"*. This is probably most evident in *teams*, which, through appropriate synergy, promise a collective outcome (i.e., team performance) that is superior than the simple addition of what each individual team member could achieve (i.e., individual productivity). For example, in professional sports (e.g., NBA), the peak performance of a grass-root team is often attributed to the harmonic teamwork between the team players rather than the individual player's capability. Beyond teams, the *part-whole* relationship also routinely finds itself in other disciplines, ranging from crowdsourcing (e.g., Community-based Question Answering (CQA) sites [16]), to reliability assessment of a networked system of components [17].

From the algorithmic perspective, an interesting problem is to predict the outcome of the whole and/or parts [18]. In organizational teams, it is critical to appraise the individual performance, its contribution to the team outcome as well as the team's overall performance [19]. Despite much progress has been made, the existing work either develop separate models for predicting the outcome of whole and parts without explicitly utilizing the part-whole relationship [14], [15], or implicitly assume the outcome of the whole is a *linear* sum of the outcome of the parts [16], which might oversimplify the complicated part-whole relationships (e.g., non-linearity). The key to address these limitations largely lies in the answers to the following questions, i.e., to what extent does the outcome of parts (e.g., individual productivity) and that of the whole (e.g., team performance) correlated, beyond the existing linear, independency assumption? How can we leverage such potentially non-linear and interdependent 'coupling' effect to mutually improve the prediction of the outcome of the whole and parts collectively? The challenges come as two-folds. First (*Modeling Challenge*), the relationship between the parts outcome and whole outcome might be complicated, beyond the simple addition or linear combination. Moreover, the composing parts of the whole might not be independent with each other. In a networked system, the composing parts are connected with each other via an underlying network. Such part-part interdependency could have a profound impact on both the part outcome correlation as well as each part's contribution to the whole outcome. Second (*Algorithmic Challenge*), the complicated part-whole relationship (i.e., non-linearity and interdependency) also poses an algorithmic challenge, as it will inevitably increase the complexity of the corresponding optimization problem.

To address these challenges, a joint predictive model named *PAROLE* is proposed to simultaneously and mutually predict the part and whole outcomes [20]. First, *model generality*, the proposed model is flexible in admitting a variety of linear as well as non-linear relationships between the parts and whole outcomes, including *maximum aggregation, linear aggregation, sparse aggregation, ordered sparse aggregation* and *robust aggregation*. Moreover, it is able to characterize part-part interdependency via a graph-based regularization, which encourages the tightly connected parts to share similar outcomes as well as have similar effect on the whole outcome. Second, *algorithm efficacy*, the authors propose an effective and efficient block coordinate descent optimization algorithm, which converges to the coordinate-wise optimum with a linear complexity.

IV. Team Performance Optimization

A. Team Formation

Team formation studies the problem of assembling a team of people to work on a project. The first work that studies team formation in the context of social networks finds a team of experts who possess the desired skills and have strong team cohesion to ensure the team success [21]. In particular, they define two communication cost based on the diameter as well as the minimum spanning tree of the induced team subgraph. Since the corresponding optimization problems are NP-complete, they devise approximation algorithms by exploiting the relationship to Multiple-Choice Cover and Group Steiner Tree problems. As follow-up work, Anagnostopoulos et al [22] study forming teams to accommodate a sequence of tasks arriving in an online fashion. Rangapuram et al [23] allow incorporating many realistic requirements (e.g., inclusion of a designated team leader) into team formation based on a generalization of the densest subgraph problem. Beyond that, minimizing the tensions among the team members is considered [24]. With the presence of the underlying social network, the set cover problem is complicated by the goal of lowering the communication cost at the same time. Cao et al [25] develop an interactive group mining system that allows users to efficiently explore the network data and from which to progressively select and replace candidate members to form a team. Bogdanov et al [26] study how to extract a diversified group pulled from strong cliques given a network; this ensures that the group is both comprehensive and representative of the whole network. Cummings and Kiesler [27] find that prior working experience is the best predictor of collaborative tie strength. To provide insights into designs of online communities and organizations, the systematic differences in appropriating social softwares among different online enterprise communities is analyzed in [28]. The patterns of informal networks and communication in distributed global software teams using social network analysis is also investigated in [29]. Specific communication structures are proven critical to new product development delivery performance and quality [30]. To assess the skills of players and teams in online multi-player games and team-based sports, "team chemistry" is also accounted for in [31], [32].

B. Team Member Replacement

The churn of team members is a common problem across many application domains. For example, an employee in a software or sales team might decide to leave the organization and/or be assigned to a new tasks. The loss of the key member (i.e., the irreplaceable) might bring the catastrophic consequence to the team performance. *How can we find the best alternate (e.g., from the other members within the organization), when a team member becomes unavailable?* Despite the frequency with which people leave a team before a project/task is complete and the resulting disruption [33], replacements are often found opportunistically and are not necessarily optimal.

It is conjectured that there will be less disruption when the team member who leaves is replaced with someone with similar relationships with the other team members. This conjecture is inspired by some recent research which shows that team members prefer to work with people they have worked with before [34] and that distributed teams perform better when members know each other [27]. Furthermore, research has shown that specific communication patterns amongst team members are critical for performance [30]. Thus, in addition to factors such as skill level, maintaining the same or better level of familiarity and communication amongst team members before and after someone leaves should reduce the impact of the departure. In other words, for team member replacement, the similarity between individuals should be measured in the context of the team itself. More specifically, a good team member replacement should meet the following two requirements. First (*skill matching*), the new member should bring a similar skill set as the current team member to be replaced. Second (*structure matching*), the new member should have a similar network structure as the current team member in connecting the rest of the team members.

Armed with this conjecture, *Team Member Replacement* problem is formally defined by the notation of graph similarity/kernel [35], [36]. By modeling the team as a labeled graph, the graph kernel provides a natural way to capture both the skill and structure match as well as the interaction of both. However, for a network with n individuals, a straightforward method would require $O(n)$ graph kernel computations for one team member replacement, which is computationally intractable. For example, for the *DBLP* dataset with almost 1M users (i.e., $n \approx 1,000,000$), the authors find that it would take 6,388s to find one replacement for a team of size 10. To address the computational challenges, they propose a family of fast algorithms by carefully designing the pruning strategies and exploring the smoothness between the existing and the new teams. From their extensive experimental evaluations, they find that (1) by encoding both the skill and structural matching, it leads to a much better replacement result. Compared with the best alternative choices, they achieve 27% and 24% *net increase* in average recall and precision, respectively; (2) the fast algorithms are orders of magnitude faster and scale *sub-linearly*. For example, their pruning strategy alone leads up to 1,709× speed-up, without sacrificing any accuracy.

C. Team Enhancement

Different from *Team Member Replacement*, *Team Refinement* considers refining a team by replacing one member with another with the desired skill sets and communication connections. In the above two problems, the team size remains the same. In *Team Expansion*, we want to expand the team by adding a member with certain skill sets and communication structure. For instance, a software project team wants to develop a new feature of natural language search and a new member with Natural Language Processing (NLP) skill will be recruited. On the contrary, in *Team Shrinkage*, the size of a team needs to be reduced in response to new challenge such as a shortage of the available resource (e.g., a budget cut). In all cases, the resulting disruption [33] should be minimized.

By careful inspection, Li et al. [36] identify the problem similarity between *Team Refinement*, *Team Expansion* and *Team Replacement* and propose these problems can be formulated in a way to share common technical solutions. In *Team Refinement*, one team member is edited to a desired skill and network structure configuration. Since such edited member might not exist in the rest of the network, they call it a 'virtual member'. By replacing this 'virtual member' as in *Team Replacement*, they can solve *Team Refinement*. Similarly, in *Team Expansion*, the desired new member might also be a 'virtual member'. After adding this 'virtual member' to the current team and then replacing the 'virtual member', they can solve *Team Expansion*. They propose to reduce the disruption induced by the team alteration by maintaining the team-level similarity (between the original and the new teams), which includes skill similarity as well as structural similarity.

D. Interactive Visualization System

A system called *TeamOPT* (http://team-net-work.org/) is developed to assist users in optimizing the team performance interactively to support the changes to a team [37] (See Fig. 2 for an example). *TeamOPT* takes as input a large network of individuals (e.g., co-author network of researchers) and is able to assist users in assembling a team with specific requirements and optimizing the team in response to the changes made to the team. To the best of our knowledge, this is the first system specializing in forming and optimizing teams with the following key features. First (*effectiveness*), they carefully identify the design objectives and develop effective algorithms with the key technique of graph kernels. Compared with other competitors, their algorithm can achieve the highest precision and recall in finding the best team member candidate. Second (*interaction*), they design fast solutions to their algorithms, enabling an interactive user experience with users' feedback

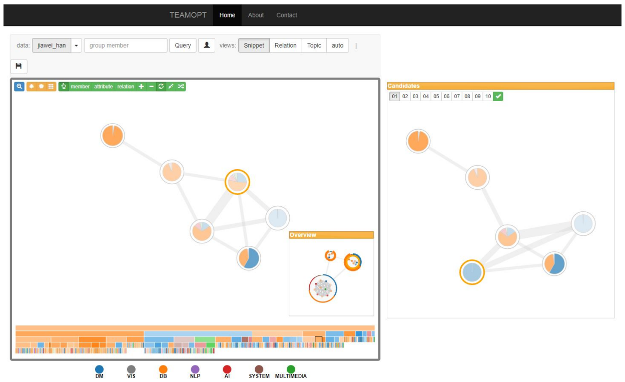


Fig. 2. A snapshot of the *TeamOPT* interactive visualization system.

in the loop. Third (*deployment*), they build the system with HTML5, Javascript, D3.js (front-end) and Python CGI (back-end).

V. Future Directions

As an emerging field, the network science of teams is still in its early stage and remains an active area of exploration. Future directions include modeling the hierarchical structure within organizations by extending the *PAROLE* model and modeling the heterogeneous goals among the team members. In the team optimization work, one implicit assumption is that the original team is performing well and maintaining the similarity with the original team can promise a similar high performance. We want to point out that when the assumption does not hold, one can leverage the actual or predicted future performance as feedback to guide the team optimization process, using advanced reinforcement learning techniques. Since team operations often involve important staffing decisions, it is critical to have team performance prediction and optimization to be explainable to the end users [38], [39].

References

[1] J. R. Hackman and N. Katz, *Group behavior and performance*. New York: Wiley, 2010, pp. 1208–1251.

[2] M. Haas and M. Mortensen, “The secrets of great teamwork.” *Harvard business review*, vol. 94, no. 6, pp. 70–6, 2016.

[3] C. Spearman, “” general intelligence,” objectively determined and measured,” *The American Journal of Psychology*, vol. 15, no. 2, pp. 201–292, 1904.

[4] A. W. Woolley, C. F. Chabris, A. Pentland, N. Hashmi, and T. W. Malone, “Evidence for a collective intelligence factor in the performance of human groups,” *science*, vol. 330, no. 6004, pp. 686–688, 2010.

[5] Y. J. Kim, D. Engel, A. W. Woolley, J. Y.-T. Lin, N. McArthur, and T. W. Malone, “What makes a strong team?: Using collective intelligence to predict team performance in league of legends,” in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, 2017, pp. 2316–2329.

[6] J. Duch, J. S. Waitzman, and L. A. N. Amaral, “Quantifying the performance of individual players in a team activity,” *PloS one*, vol. 5, no. 6, p. e10937, 2010.

[7] T. U. Grund, “Network structure and team performance: The case of english premier league soccer teams,” *Social Networks*, vol. 34, no. 4, pp. 682–690, 2012.

[8] F. Thung, T. F. Bissyande, D. Lo, and L. Jiang, “Network structure of social coding in github,” in *Software maintenance and reengineering (csmr), 2013 17th european conference on*. IEEE, 2013, pp. 323–326.

[9] S. Wuchty, B. F. Jones, and B. Uzzi, “The increasing dominance of teams in production of knowledge,” *Science*, vol. 316, no. 5827, pp. 1036–1039, 2007.

[10] B. Uzzi, S. Mukherjee, M. Stringer, and B. Jones, “Atypical combinations and scientific impact,” *Science*, vol. 342, no. 6157, pp. 468–472, 2013.

[11] D. Wang, C. Song, and A.-L. Barabási, “Quantifying long-term scientific impact,” *Science*, vol. 342, no. 6154, pp. 127–132, 2013.

[12] C. Castillo, D. Donato, and A. Gionis, “Estimating number of citations using author reputation,” in *String processing and information retrieval*. Springer, 2007, pp. 107–117.

[13] R. Yan, J. Tang, X. Liu, D. Shan, and X. Li, “Citation count prediction: learning to estimate future citations for literature,” in *CIKM*, 2011, pp. 1247–1252.

[14] L. Li and H. Tong, “The child is father of the man: Foresee the success at the early stage,” in *KDD*. ACM, 2015, pp. 655–664.

[15] L. Li, H. Tong, J. Tang, and W. Fan, “*iPath:* forecasting the pathway to impact,” in *SDM*. SIAM, 2016, pp. 468–476.

[16] Y. Yao, H. Tong, F. Xu, and J. Lu, “Predicting long-term impact of cqa posts: a comprehensive viewpoint,” in *KDD*. ACM, 2014, pp. 1496–1505.

[17] C. Chen, H. Tong, L. Xie, L. Ying, and Q. He, “Fascinate: Fast cross-layer dependency inference on multi-layered networks,” in *KDD*, ser. KDD ’16, 2016, pp. 765–774.

[18] B. R. Jasny and R. Stone, “Prediction and its limits,” *Science*, vol. 355, no. 6324, pp. 468–469, 2017.

[19] L. Liu and E. Zhao, “Team performance and individual performance: Example from engineering consultancy company in china,” in *2011 International Conference on Management and Service Science*, Aug 2011, pp. 1–4.

[20] L. Li, H. Tong, Y. Wang, C. Shi, N. Cao, and N. Buchler, “Is the whole greater than the sum of its parts?” in *KDD*, 2017, pp. 295–304.

[21] T. Lappas, K. Liu, and E. Terzi, “Finding a team of experts in social networks,” in *KDD*, 2009, pp. 467–476.

[22] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, “Online team formation in social networks,” in *WWW*, 2012, pp. 839–848.

[23] S. S. Rangapuram, T. Bühler, and M. Hein, “Towards realistic team formation in social networks based on densest subgraphs,” in *WWW*, 2013, pp. 1077–1088.

[24] B. Golshan and E. Terzi, “Minimizing tension in teams,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM ’17. New York, NY, USA: ACM, 2017, pp. 1707–1715.

[25] N. Cao, Y.-R. Lin, L. Li, and H. Tong, “g-Miner: Interactive visual group mining on multivariate graphs,” in *CHI*, 2015.

[26] P. Bogdanov, B. Baumer, P. Basu, A. Bar-Noy, and A. K. Singh, “As strong as the weakest link: Mining diverse cliques in weighted graphs,” in *ECML/PKDD (1)*, 2013, pp. 525–540.

[27] J. N. Cummings and S. B. Kiesler, “Who collaborates successfully?: prior experience reduces collaboration barriers in distributed interdisciplinary research,” in *CSCW*, 2008, pp. 437–446.

[28] M. Muller, K. Ehrlich, T. Matthews, A. Perer, I. Ronen, and I. Guy, “Diversity among enterprise online communities: collaborating, teaming, and innovating through social media,” in *CHI*, 2012, pp. 2815–2824.

[29] K. Chang and K. Ehrlich, “Out of sight but not out of mind?: Informal networks, communication and media use in global software teams,” in *CASCON*, 2007, pp. 86–97.

[30] M. Cataldo and K. Ehrlich, “The impact of communication structure on new product development outcomes,” in *CHI*, 2012, pp. 3081–3090.

[31] C. DeLong, L. G. Terveen, and J. Srivastava, “Teamskill and the nba: applying lessons from virtual worlds to the real-world,” in *ASONAM*, 2013, pp. 156–161.

[32] C. DeLong and J. Srivastava, “Teamskill evolved: Mixed classification schemes for team-based multi-player games,” in *PAKDD (1)*, 2012, pp. 26–37.

[33] R. Zadeh, A. D. Balakrishnan, S. B. Kiesler, and J. N. Cummings, “What’s in a move?: normal disruption and a design challenge,” in *CHI*, 2011, pp. 2897–2906.

[34] P. J. Hinds, K. M. Carley, D. Krackhardt, and D. Wholey, “Choosing work group members: Balancing similarity, competence, and familiarity,” in *Organizational Behavior and Human Decision Processes*, 2000, pp. 226–251.

[35] L. Li, H. Tong, N. Cao, K. Ehrlich, Y. Lin, and N. Buchler, “Replacing the irreplaceable: Fast algorithms for team member recommendation,” in *WWW*. ACM, 2015, pp. 636–646.

[36] ——, “Enhancing team composition in professional networks: Problem definitions and fast solutions,” *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 3, pp. 613–626, 2017.

[37] ——, “TEAMOPT: interactive team optimization in big networks,” in *CIKM*, 2016, pp. 2485–2487.

[38] L. Li, H. Tong, and H. Liu, “Towards explainable networked prediction,” in *CIKM*, 2018, pp. 1819–1822.

[39] Q. Zhou, L. Li, N. Cao, N. Buchler, and H. Tong, “Extra: explaining team recommendation in networks,” in *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, 2018, pp. 492–493.

Zeroth-Order Optimization and Its Application to Adversarial Machine Learning

Sijia Liu and Pin-Yu Chen

***Abstract*—Many big data problems deal with complex data generating processes that cannot be described by analytical forms but can provide function evaluations, such as measurements from physical environments or predictions from deployed machine learning models. These types of problems fall into zeroth-order (gradient-free) optimization with respect to black-box models. In this paper, we provide a comprehensive introduction to recent advances in zeroth-order (ZO) optimization methods in both theory and applications. On the theory side, we will elaborate on ZO gradient estimation and the convergence rate of various ZO algorithms. The existing studies suggest that ZO algorithms typically agree with the iteration complexity of first-order algorithms up to a small-degree polynomial of the problem size. On the application side, we will delve into applications of ZO algorithms on studying the robustness of deep neural networks against adversarial perturbations. In particular, we will illustrate how to formulate the design of black-box adversarial attacks as a ZO optimization problem and how adversarial attacks can benefit from advanced ZO optimization techniques, such as providing query-efficient approaches to generating adversarial examples from a black-box image classifier.**

***Index Terms*—Zeroth-order optimization, adversarial machine learning, black-box adversarial example, gradient estimation.**

I. Introduction

ZEROTH-order (ZO) optimization is increasingly embraced for solving big data and machine learning problems when explicit expressions of the gradients are difficult or infeasible to obtain. It achieves gradient-free optimization by approximating the full gradient via efficient gradient estimators. One recent application of particular interest is to generate prediction-evasive adversarial examples using only the input-output correspondence of the target machine learning model, e.g., deep neural networks (DNNs) [1]–[4]. Additional classes of applications include network control and management with time-varying constraints and limited computation capacity [5], [6], parameter inference of black-box systems [7]–[9], and bandit optimization in which a player receives partial feedback in terms of loss function values revealed by her adversary [10], [11].

Spurred by application demands for ZO optimization, many types of ZO algorithms were developed for convex and nonconvex optimization. In these algorithms, a full gradient is typically approximated using either a one-point or a two-point gradient estimator, where the former acquires a gradient estimate by querying the (black-box) objective function $f(\mathbf{x})$ at a single random location close to $\mathbf{x}$ [10], [12], and the latter computes a finite difference using two random function queries [13], [14]. Compared to the one-point gradient estimator, the *two-point* gradient estimator has a lower variance and thus improves the complexity bounds of ZO algorithms.

Despite the meteoric rise of two-point based ZO algorithms, most of the work is restricted to convex problems [6], [11], [15]–[18]. For example, a ZO mirror descent algorithm proposed by [15] has an exact rate $O(\sqrt{d}/\sqrt{T})$, where d is the number of optimization variables, and T is the number of iterations. The same rate is obtained by bandit convex optimization [11] and ZO online alternating direction method of multipliers [6].

In contrast to the convex setting, non-convex ZO optimization introduces a large amount of recent attention [8], [14], [19]–[21]. Different from convex optimization, the stationary condition is used to measure the convergence of nonconvex methods. In [14], the ZO gradient descent (ZO-GD) algorithm was proposed for deterministic nonconvex programming, which yields $O(d/T)$ convergence rate. A stochastic version of ZO-GD (namely, ZO-SGD) studied in [19] achieves the rate of $O(\sqrt{d}/\sqrt{T})$. In [20], a ZO distributed algorithm was developed for multi-agent optimization, leading to $O(1/T + d/q)$ convergence rate. Here q is the number of random directions used to construct a gradient estimate. In [8], an asynchronous ZO stochastic coordinate descent (ZO-SCD) was derived for parallel optimization and achieved the rate of $O(\sqrt{d}/\sqrt{T})$. In [9], [21], a stochastic variance reduced technique was used to achieve the improved convergence rate of $O(d/T)$.

Current studies suggested that ZO algorithms typically agree with the iteration complexity of first-order algorithms up to a small-degree polynomial of the problem size. In this paper, we will investigate how (two-point) random gradient estimate fits into ZO optimization. We will also survey the convergence rate of existing ZO optimization algorithms. Lastly, we will delve into applications of ZO algorithms to study the robustness of deep neural networks against adversarial perturbations.

II. Random Gradient Estimation via Zeroth-Order Oracle

We consider a finite-sum optimization problem of the form

$$\underset{\mathbf{x}\in\mathcal{C}}{\text{minimize}} \quad f(\mathbf{x}) := (1/n)\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the optimization variable, $\mathcal{C} \in \mathbb{R}^d$ is a convex constraint set, and $\{f_i(\mathbf{x})\}$ are n component functions (not necessarily convex). In (1), if $\mathcal{C} = \mathbb{R}^d$, then we study an unconstrained finite-sum problem.

Compared to first-order optimization, ZO optimization requires to approximate the first-order gradient of $f(\mathbf{x})$ only

The authors are from the MIT-IBM Watson AI Lab, IBM Research, USA. Corresponding Author e-mail: ({sijia.liu, pin-yu.chen}@ibm.com).

through function values. Given a component function f_i, a two-point based average random gradient estimator $\hat{\nabla} f_i(\mathbf{x})$ is defined by [9], [11], [14], [16]

$$\hat{\nabla} f_i(\mathbf{x}) = \frac{d}{q} \sum_{j=1}^{q} \left[\frac{f_i(\mathbf{x} + \mu \mathbf{u}_j) - f_i(\mathbf{x})}{2\mu} \mathbf{u}_j \right], \qquad (2)$$

where d is the number of optimization variables, $\mu > 0$ is a smoothing parameter, and $\{\mathbf{u}_j\}_{j=1}^q$ are i.i.d. random directions drawn from a uniform distribution over a unit sphere [11], [16]. Notably, the random direction vector $\mathbf{u}_j$ can also be drawn from the standard Gaussian distribution [14], [20], [22]. However, we argue that the uniform distribution could be more useful in practice since it is defined in a *bounded* space rather than the *whole* real space required for Gaussian. In (2), the larger q is, the smaller the variance of ZO gradient estimate is. Also, the gradient estimate (2) requires $(q+1)$ function queries. Clearly, the parameter q plays a trade-off between the variance of ZO gradient estimate and the function query complexity.

We highlight that unlike the first-order stochastic gradient estimate, the ZO gradient estimate (2) is a biased approximation to the true gradient of f_i. Instead, it becomes unbiased to the gradient of the randomized smoothing version of f_i [15], [16],

$$f_{i,\mu}(\mathbf{x}) = \mathbb{E}_{\mathbf{v}}[f_i(\mathbf{x} + \mu \mathbf{v})], \qquad (3)$$

where $f_{i,\mu}$ is called the randomized smoothing version of f_i with smoothing parameter μ, and the random variable $\mathbf{v}$ follows a uniform distribution over the unit Euclidean ball. Although there exists a gap between a ZO gradient estimate and the true gradient of f_i, such a gap can be measured through its smoothing function.

In what follows, we derive the key statistical properties of the ZO gradient estimate (2).

Lemma 1: The ZO gradient estimate (2) yields:

1) For any $\mathbf{x} \in \mathbb{R}^d$

$$\mathbb{E}\left[\hat{\nabla} f_i(\mathbf{x})\right] = \nabla f_{i,\mu}(\mathbf{x}). \qquad (4)$$

2) Suppose that f_i has L-Lipschitz continuous gradient, then for any $\mathbf{x} \in \mathbb{R}^d$

$$\begin{aligned} &\mathbb{E}\left[\|\hat{\nabla} f_i(\mathbf{x}) - \nabla f_{i,\mu}(\mathbf{x})\|_2^2\right] \\ \leq &2\left(1 + \frac{d}{q}\right) \|\nabla f_i(\mathbf{x})\|_2^2 + \left(1 + \frac{1}{q}\right) \frac{\mu^2 L^2 d^2}{2}. \end{aligned} \qquad (5)$$

Proof: see [9, Lemma 2]. □

Lemma 1 uncovers important properties of ZO gradient estimation. First, the use of *multiple* ($q > 1$) random direction vectors $\{\mathbf{u}_j\}$ does not reduce the bias of $\hat{\nabla} f_i$ (with respect to ∇f_i) since $\hat{\nabla} f$ is unbiased only with respect to ∇f_μ. Second, the variance of the random gradient estimator is reduced as q increases. In particular, a large q mitigates the dimension (d) dependency on the second-order moment of (2). This is crucial to improve the convergence performance of ZO optimization algorithms.

III. Convergence Analysis of Zeroth-Order Optimization Algorithms

In this section, we review the existing ZO algorithms that can be used to solve problem (1) and elaborate on their convergence rates. We divide the studied algorithms into two categories for unconstrained optimization and constrained optimization, respectively. Moreover, if problem (1) is convex, we use the optimality gap $f(\mathbf{x}) - f(\mathbf{x}^*)$ to measure the convergence rate, where $\mathbf{x}^*$ is the globally optimal solution. When problem (1) is nonconvex and unconstrained, we measure the stationarity in terms of $\|\nabla f(\mathbf{x})\|_2^2$. For constrained non-convex problems, a fitting alternative of $\|\nabla f(\mathbf{x})\|_2^2$, called gradient mapping, is then used for convergence evaluation [22]–[24].

A. ZO algorithms for unconstrained optimization

1) ZO gradient descent (ZO-GD) [14]: At the kth iteration, ZO-GD updates the solution as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \hat{\nabla} f(\mathbf{x}_k), \qquad (6)$$

where $\eta_k > 0$ is learning rate, and $\hat{\nabla} f(\mathbf{x}_k) = \frac{1}{n} \sum_{i=1}^{n} \hat{\nabla} f_i(\mathbf{x}_k)$.

2) ZO stochastic gradient descent (ZO-SGD) [19] :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \left(\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \hat{\nabla} f_i(\mathbf{x}_k) \right), \qquad (7)$$

where $\mathcal{B}$ is a mini-batch of size $|\mathcal{B}|$, and $\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \hat{\nabla} f_i(\mathbf{x}_k)$ is an estimate of stochastic gradient under mini-batch $\mathcal{B}$.

3) ZO stochastic coordinate descent (ZO-SCD) [8] :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \tilde{\nabla} f_{i_k}(\mathbf{x}_k), \qquad (8)$$

where i_k is a component function index randomly picked from $[n] := \{1, 2, \ldots, n\}$, and $\tilde{\nabla} f_{i_k}(\mathbf{x}_k)$ is an estimate of a block coordinate stochastic gradient given by $\frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \left(\frac{d}{2\mu} \left(f_{i_k}(\mathbf{x}_k + \mu \mathbf{e}_j) - f_{i_k}(\mathbf{x}_k - \mu \mathbf{e}_j) \right) \mathbf{e}_j \right)$. Here $\mathcal{S}$ is a mini-batch of coordinates randomly selected from $[d]$.

4) ZO sign-based stochastic gradient descent (ZO-signSGD) [25] :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathrm{sign} \left(\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \hat{\nabla} f_i(\mathbf{x}_k) \right), \qquad (9)$$

where $\mathrm{sign}(\cdot)$ takes element-wise signs of $\mathbf{x}$. It is shown in [25] that the convergence of ZO-signSGD can be measured via $\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2]$, a stricter criterion than $\mathbb{E}[\|\nabla f(\mathbf{x}_T)\|_2^2]$.

For ease of comparison, we do not incorporate variance reduced versions of ZO algorithms, e,g., ZO-SVRG and ZO-SVRC, which are ZO stochastic variance reduced gradient/coordinate descent algorithms in [9], [21]. That is because those algorithms require extra query complexity in order to achieve better convergence rates.

B. Constrained optimization

1) ZO stochastic mirror descent (ZO-SMD) [15] :

$$\mathbf{x}_{k+1} = \underset{\mathbf{x} \in \mathcal{C}}{\arg\min} \left\{ \langle \hat{\mathbf{g}}_k, \mathbf{x} \rangle + \frac{1}{\eta_k} \|\mathbf{x} - \mathbf{x}_k\|_2^2 \right\}, \qquad (10)$$

where for east of notation, let $\hat{\mathbf{g}}_k = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \hat{\nabla} f_i(\mathbf{x}_k)$.

TABLE I: Summary of convergence rate and query complexity of various ZO algorithms.

Method	Problem setting	Gradient estimator	Smoothing parameter μ	Convergence rate	Query complexity (T iterations)
ZO-GD [14]	nonconvex, unconstrained	GauSGE*	$O\left(\frac{1}{\sqrt{dT}}\right)$	$\mathbb{E}[\Vert\nabla f(\mathbf{x}_T)\Vert_2^2] = O\left(\frac{d}{T}\right)$	$O\left(\vert\mathcal{B}\vert qT\right)$
ZO-SGD [19]	nonconvex, unconstrained	GauSGE	$O\left(\frac{1}{d\sqrt{T}}\right)$	$\mathbb{E}[\Vert\nabla f(\mathbf{x}_T)\Vert_2^2] = O\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$	$O\left(\vert\mathcal{B}\vert qT\right)$
ZO-SCD [8]	nonconvex, unconstrained	CooGE	$O\left(\frac{1}{\sqrt{T}} + \min\{\frac{1}{(dT)^{-1/4}}, \frac{1}{\sqrt{d}}\}\right)$	$\mathbb{E}[\Vert\nabla f(\mathbf{x}_T)\Vert_2^2] = O\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$	$O\left(\vert\mathcal{B}\vert\vert\mathcal{S}\vert T\right)$
ZO-signSGD [25]	nonconvex, unconstrained	GauSGE	$O\left(\frac{1}{\sqrt{dT}}\right)$	$\mathbb{E}[\Vert\nabla f(\mathbf{x}_T)\Vert_2] = O\left(\frac{\sqrt{d}}{\sqrt{T}} + \frac{\sqrt{d}}{\sqrt{\vert\mathbf{B}\vert}} + \frac{d}{\sqrt{q\vert\mathcal{B}\vert}}\right)$	$O\left(\vert\mathcal{B}\vert qT\right)$
ZO-SVRG [9]	nonconvex, unconstrained	UniSGE*	$O\left(\frac{1}{\sqrt{dT}}\right)$	$\mathbb{E}[\Vert\nabla f(\mathbf{x}_T)\Vert_2^2] = O\left(\frac{d}{T} + \frac{1}{\sqrt{\vert\mathbf{B}\vert}}\right)$	$O\left(qnS + q\vert\mathcal{B}\vert Sm\right)$, $T = Sm^{**}$
ZO-SMD [15]	convex, constrained	GauSGE/UniSGE	$O\left(\frac{1}{dt}\right)$	$\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] = O\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$	$O(T)$
ZO-PSGD [26]	nonconvex, constrained	UniSGE	$O\left(\frac{1}{\sqrt{dq\vert\mathcal{B}\vert}}\right)$	$\mathbb{E}[\Vert\nabla f(\mathbf{x}_T)\Vert_2^2] = O\left(\frac{1}{\sqrt{T}} + \frac{d+q}{bq}\right)$	$O\left(\vert\mathcal{B}\vert qT\right)$
ZO-FW [27]	nonconvex, constrained	GauSGE/UniSGE	$O\left(\frac{1}{d^{1.5}t^{1/3}}\right)$	$\mathbb{E}[\Vert\nabla f(\mathbf{x}_T)\Vert_2^2] = O\left(\frac{(d/q)^{1/3}}{T^{1/4}}\right)$	$O\left(qT\right)$
ZO-ProxSGD [22]	nonconvex, composite***	GauSGE	$O\left(\frac{1}{\sqrt{dT}}\right)$	$\mathbb{E}[\Vert\nabla f(\mathbf{x}_T)\Vert_2^2] = O\left(\frac{d}{\vert\mathcal{B}\vert qT} + \frac{d^2}{\vert\mathcal{B}\vert qT} + \frac{d}{\vert\mathcal{B}\vert q}\right)$	$O\left(\vert\mathcal{B}\vert qT\right)$
ZO-OADMM [6]	convex, composite	GauSGE/UniSGE	$O\left(\frac{1}{d^{1.5}t}\right)$	$\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] = O\left(\frac{\sqrt{d}}{\sqrt{T\vert\mathcal{B}\vert q}}\right)$	$O\left(\vert\mathcal{B}\vert qT\right)$

[*] GauSGE and UniSGE represents the ZO gradient estimator using random direction vectors generated from the standard normal distribution and the uniform distribution over a unit sphere, respectively.

[**] ZO-SVRG contains two iteration loops, where the number of outer iterations is S and the number of inner iterations is m.

[***] Composite optimization can handle smooth + nonsmooth objective functions.

2) ZO projected stochastic gradient descent (ZO-PSGD) [26] :

$$\mathbf{x}_{k+1} = \Pi_{\mathcal{C}}\left[\mathbf{x}_k - \eta_k \hat{\mathbf{g}}_k\right], \tag{11}$$

where $\Pi_{\mathcal{C}}$ denotes the projection operator with respect to $\mathcal{C}$, i.e., $\Pi_{\mathcal{C}}(\mathbf{a}) = \arg\min_{\mathbf{z}\in\mathcal{C}} \|\mathbf{z} - \mathbf{a}\|^2$. We remark that ZO-PSGD can be regarded as a special case of ZO proximal stochastic gradient descent (ZO-ProxSGD), which is proposed to solve constrained composite optimization problems [22]. However, the complexity of ZO-ProxSGD is dominated by the computation of the proximal operation with respect to all nonsmooth regularization functions. To overcome this issue, reference [6] developed a ZO online alternating direction method of multipliers (ZO-OADMM) algorithm, which can split the original complex optimization problem into a sequence of easily-solved subproblems in a flexible manner.

3) ZO Frank-Wolfe (ZO-FW) [27] : The ZO Frank-Wolfe algorithm calls the following linear minimization oracle (LMO) at each iteration

$$\begin{aligned} \mathbf{v}_k &= \arg\min_{\mathbf{x}\in\mathcal{C}} \langle \hat{\mathbf{g}}_k, \mathbf{x}\rangle \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \eta_k(\mathbf{v}_k - \mathbf{x}_k) \end{aligned} \tag{12}$$

where the ZO gradient estimate $\hat{\mathbf{g}}_k$ has been defined in (10). We note that the LMO is equivalent to the minimization of the first-order Taylor expansion of f at point $\mathbf{x}_k$ using the ZO gradient estimate $\hat{\mathbf{g}}_k$.

As a concluding remark, we summarize the settings, ZO gradient estimators, and convergence rates of various ZO algorithms in Table I.

IV. Black-Box Adversarial Attacks: An ZO Optimization Perspective

In this section, we will illustrate how to formulate black-box adversarial attacks as a ZO optimization problem and how adversarial attacks can benefit from advanced ZO optimization techniques, such as providing query-efficient approaches to generating adversarial examples from a black-box image classifier.

Generally speaking, given a natural input $\mathbf{x}_0$ to a machine learning model, its adversarial example $\mathbf{x}$ refers to a modified input which is (semantically) close to $\mathbf{x}_0$ but the model outputs of $\mathbf{x}$ and $\mathbf{x}_0$ are drastically different, e.g., classifying $\mathbf{x}_0$ as a label t_0 but classifying $\mathbf{x}$ as another label $t \neq t_0$. The adversarial modification can be accomplished by considering the additive perturbation model $\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\delta}$, and the level of distortion is often measured by the ℓ_p norm ($p \geq 1$) of the perturbation $\boldsymbol{\delta}$, particularly the ℓ_1, ℓ_2 and ℓ_∞ norms [28]–[30]. When the distortion is small, the adversarial perturbation is visually imperceptible but can cause the target machine learning model to misbehave, resulting in increasing concerns in safety-critical and cost-sensitive applications, as well as new challenges in training robust machine learning models.

Typically, the adversarial perturbations are crafted in the "white-box" setting, where the adversary has full access to the target model such as model parameters and neural network structures. Take neural network classifiers as an example, adversarial perturbations for misclassification can be found by performing back-propagation through network layers from the model output to the model input. With some designed attack loss function (e.g., cross entropy), back-propagation provides the direction of making the perturbed input adversarial and can be applied successively to perturbed inputs.

While in the white-box setting crafting adversarial examples are shown to be plausible in many machine learning tasks, spanning from image classification [31], speech recognition [32], machine translation [33], image captioning [34], text sentiment analysis [35] to sparse regression [36], the need for requiring back-propagation of the target model renders white-box adversarial attacks less practical when attacking a deployed machine learning service, such as Google Cloud Vision API[1] and Clarifai.com[2]. In this case, one only has access to the model output (e.g., class prediction scores) of a queried input but is completely agnostic about the target model, which is known as the black-box attack setting. The target model can be a neural network, a support vector machine, a decision tree,

[1] https://cloud.google.com/vision

[2] https://clarifai.com

or any other classifier. One question that naturally arises is: *How can we generate adversarial examples in the black-box setting?* Notably, this problem setup of black-box adversarial attacks fits into the framework of ZO optimization. One can formulate the process of finding an adversarial example of a black-box model as a ZO optimization problem, where the objective function is associated with the model output and the gradient is infeasible to obtain (e.g., back-propagation is inadmissible in the black-box setting).

Without lose of generality, we denote a target machine learning model as a classification function $F : [0,1]^d \mapsto \mathbb{R}^K$ that takes a d-dimensional scaled data sample as its input and yields a vector of prediction scores of all K image classes, such as the prediction probabilities for each class. We further consider the case of applying an entry-wise monotonic transformation $M(F)$ to the output of F for black-box attacks, since monotonic transformation preserves the ranking of the class predictions and can alleviate the problem of large score variation in F (e.g., probability to log probability). As an illustration, we use the black-box targeted attack loss function proposed in [3], which aims to minimize the following objective function

$$\text{minimize}_{\boldsymbol{\delta}:\mathbf{x}_0+\boldsymbol{\delta}\in[0,1]^d} \ \|\boldsymbol{\delta}\|_2^2 + \lambda \cdot \text{Loss}(\mathbf{x}, M(F(\mathbf{x})), t), \tag{13}$$

where $\|\boldsymbol{\delta}\|_2^2$ measures the distortion between $\mathbf{x}$ and $\mathbf{x}_0$ in the squared ℓ_2 norm and $\text{Loss}(\cdot)$ is an attack objective reflecting the likelihood of predicting $t = \arg\max_{k\in\{1,\dots,K\}}[M(F(\mathbf{x}))]_k$, λ is a regularization coefficient, and the constraint $\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\delta} \in [0,1]^d$ confines the adversarial example $\mathbf{x}$ to the valid sample space. Specifically, the Loss term is defined as

$$\text{Loss} = \max\{\max_{j\neq t} \log[F(\mathbf{x})]_j - \log[F(\mathbf{x})]_t\}, -\kappa\}, \tag{14}$$

where the monotonic transformation $M(\cdot) = \log(\cdot)$ is applied to the model output $F(\cdot)$, the constant parameter $\kappa \geq 0$ controls the gap between the confidence of target class label $\log[F(\mathbf{x})]_t$ and the second highest class label $\max_{j\neq t}\log[F(\mathbf{x})]_j$, and the hinge-like term $\max\{\cdot, -\kappa\}$ ensures this term is a constant $-\kappa$ once $\log[F(\mathbf{x})]_t - \max_{j\neq t}\log[F(\mathbf{x})]_j \geq \kappa$. For untargeted attacks that aim at classifying $\mathbf{x}$ as any label other than the original top-1 label t_0 of $\mathbf{x}_0$, the loss term can be defined as

$$\text{Loss} = \max\{\log[F(\mathbf{x})]_{t_0} - \max_{j\neq t}\log[F(\mathbf{x})]_j, -\kappa\}. \tag{15}$$

The constraint $\mathbf{x}_0 + \boldsymbol{\delta} \in [0,1]^d$ in (13) can be eliminated via change-of-variable (e.g., using tanh transformation) such that the black-box attack formulation becomes an unconstrained zeroth-order optimization problem.

Here we discuss two ZO optimization based black-box adversarial attacks on the Inception-v3 model [37] trained on ImageNet: (i) the ZOO attack [3] and (ii) the AutoZOOM attack [4]. The ZOO attack adopts random block coordinate descent for solving (13), whereas AutoZOOM adopts the two-point based average random gradient estimator as in (2) and uses dimension reduction on the perturbation $\boldsymbol{\delta}$ (either an off-line trained autoencoder or a bilinear resizer) to improve the efficiency in model query. The parameter settings and the displayed images are adopted from these two papers.

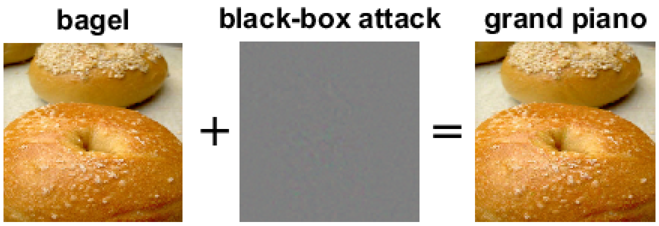


Fig. 1: Targeted black-box adversarial example (original class: bagel; targeted class: grand piano) using the ZOO attack [3] on the black-box Inception-v3 model. Left: original image ($\mathbf{x}_0$). Middle: adversarial perturbation ($\boldsymbol{\delta}$). Right: adversarial example ($\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\delta}$).

Fig. 1 shows an adversarial bagel image with a target label “grand piano” using the ZOO attack. It can be observed that the adversarial perturbation is indeed visually imperceptible but will cause the resulting adversarial example to be misclassified as grand piano. Notably, it has been shown in [3] that even without using back-propagation, the distortion level of black-box adversarial attacks can be comparable to that of white-box adversarial attacks, suggesting the effectiveness of ZO optimization. Intuitively, in the context of black-box adversarial attacks, the success of ZO optimization with gradient estimates can be anticipated as it is performing a “psuedo back-propagation” of the target model. Furthermore, its reliable attack performance is assured by the convergence analysis. Fig. 2 compares the performance of the ZOO attack and the AutoZOOM attack on the same image and target label. With the use of two point based average random gradient estimator in AutoZOOM instead of the coordinate-wise gradient estimator in ZOO, the AutoZOOM attack significantly reduces the number of queries (about 83%) required to generate a visually similar adversarial bagel image from the black-box Inception-v3 model. The remarkable improvement in query efficiency is consistent with the query complexity analysis between ZO-SCD and ZO-SGD as discussed in the previous sections and Table I. It is also worth noting that even in the stringent attacking scenario where the target black-box classifier only outputs the top-1 prediction label of a queried input, ZO optimization with some additional objective function smoothing techniques can still be used to craft adversarial examples [38], [39].

V. Conclusion

This paper provides a systematic and comprehensive overview of zeroth-order (ZO) optimization, which only requires function evaluations to solve for a finite-sum minimization problem with optionally convex set constraints. We discuss several gradient estimation based ZO optimization methods and compare their performance in terms of convergence rate and query complexity. As a motivating example, we highlight how ZO optimization can be used to craft adversarial examples of a black-box machine learning model in an efficient and principled manner.

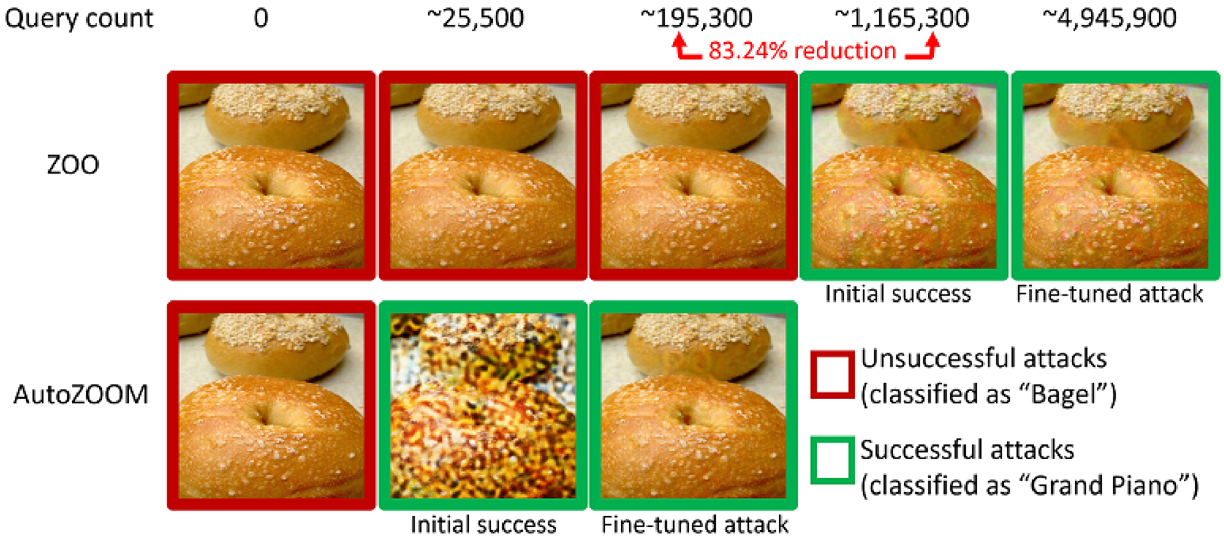


Fig. 2: Comparison of ZOO and AutoZOOM black-box adversarial attacks. With the use of the two point based average random gradient estimator in AutoZOOM instead of the coordinate-wise gradient estimator in ZOO, AutoZOOM significantly reduces the number of queries required to generate a visually similar adversarial bagel image from the black-box Inception-v3 model.

ACKNOWLEDGMENT

The authors thank the MIT-IBM Watson AI Lab which fully for fully supporting this research. The authors also would like to thank Mingyi Hong for the helpful discussion on zeroth-order optimization.

REFERENCES

[1] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017, pp. 506–519.

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[3] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 15–26.

[4] C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng, "Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," *arXiv preprint arXiv:1805.11770*, 2018.

[5] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic IoT management," *IEEE Internet of Things Journal*, 2018.

[6] S. Liu, J. Chen, P.-Y. Chen, and A. O. Hero, "Zeroth-order online ADMM: Convergence analysis and applications," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, vol. 84, April 2018, pp. 288–297.

[7] M. C. Fu, "Optimization for simulation: Theory vs. practice," *INFORMS Journal on Computing*, vol. 14, no. 3, pp. 192–215, 2002.

[8] X. Lian, H. Zhang, C.-J. Hsieh, Y. Huang, and J. Liu, "A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order," in *Advances in Neural Information Processing Systems*, 2016, pp. 3054–3062.

[9] S. Liu, B. Kailkhura, P.-Y. Chen, P. Ting, S. Chang, and L. Amini, "Zeroth-order stochastic variance reduction for nonconvex optimization," *Advances in Neural InformationProcessing Systems*, 2018.

[10] O. Shamir, "On the complexity of bandit and derivative-free stochastic convex optimization," in *Conference on Learning Theory*, 2013, pp. 3–24.

[11] ——, "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback," *Journal of Machine Learning Research*, vol. 18, no. 52, pp. 1–11, 2017.

[12] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, 2005, pp. 385–394.

[13] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback," in *COLT*, 2010, pp. 28–40.

[14] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 2, no. 17, pp. 527–566, 2015.

[15] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2788–2806, 2015.

[16] X. Gao, B. Jiang, and S. Zhang, "On the information-adaptive variants of the ADMM: an iteration complexity perspective," *Optimization Online*, vol. 12, 2014.

[17] P. Dvurechensky, A. Gasnikov, and E. Gorbunov, "An accelerated method for derivative-free smooth stochastic convex optimization," *arXiv preprint arXiv:1802.09022*, 2018.

[18] Y. Wang, S. Du, S. Balakrishnan, and A. Singh, "Stochastic zeroth-order optimization in high dimensions," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, vol. 84. PMLR, April 2018, pp. 1356–1365.

[19] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[20] D. Hajinezhad, M. Hong, and A. Garcia, "Zeroth order nonconvex multi-agent optimization over networks," *arXiv preprint arXiv:1710.09997*, 2017.

[21] B. Gu, Z. Huo, and H. Huang, "Zeroth-order asynchronous doubly stochastic algorithm with variance reduction," *arXiv preprint arXiv:1612.01425*, 2016.

[22] S. Ghadimi, G. Lan, and H. Zhang, "Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization," *Mathematical Programming*, vol. 155, no. 1-2, pp. 267–305, 2016.

[23] S. J. Reddi, S. Sra, B. Poczos, and A. J. Smola, "Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization," in *Advances in Neural Information Processing Systems*, 2016, pp. 1145–1153.

[24] E. Hazan, K. Singh, and C. Zhang, "Efficient regret minimization in non-convex games," *arXiv preprint arXiv:1708.00075*, 2017.

[25] Anonymous, "signsgd via zeroth-order oracle," in *Submitted to International Conference on Learning Representations*, 2019, under review. [Online]. Available: https://openreview.net/forum?id=BJe-DsC5Fm

[26] S. Liu, X. Li, P.-Y. Chen, B. Vinzamuri, J. Haupt, and L. Amini, "Zeroth-

order stochastic projected gradient descent for nonconvex optimization," in *GlobalSIP*. IEEE, 2018.

[27] A. K. Sahu, M. Zaheer, and S. Kar, "Towards gradient free and projection free stochastic optimization," *arXiv preprint arXiv:1810.03233*, 2018.

[28] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "EAD: elastic-net attacks to deep neural networks via adversarial examples," *AAAI*, 2018.

[29] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy*, 2017, pp. 39–57.

[30] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *ICLR*, 2017.

[31] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, "Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models," in *ECCV*, 2018, pp. 631–648.

[32] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," *arXiv preprint arXiv:1801.01944*, 2018.

[33] M. Cheng, J. Yi, H. Zhang, P.-Y. Chen, and C.-J. Hsieh, "Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples," *arXiv preprint arXiv:1803.01128*, 2018.

[34] H. Chen, H. Zhang, P.-Y. Chen, J. Yi, and C.-J. Hsieh, "Attacking visual language grounding with adversarial examples: A case study on neural image captioning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2018, pp. 2587–2597.

[35] Q. Lei, L. Wu, P.-Y. Chen, A. G. Dimakis, I. S. Dhillon, and M. Witbrock, "Discrete attacks and submodular optimization with applications to text classification," *arXiv preprint arXiv:1812.00151*, 2018.

[36] P.-Y. Chen, B. Vinzamuri, and S. Liu, "Is ordered weighted ℓ_1 regularized regression robust to adversarial perturbation? a case study on oscar," *arXiv preprint arXiv:1809.08706*, 2018.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[38] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," *ICML*, 2018.

[39] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh, "Query-efficient hard-label black-box attack: An optimization-based approach," *arXiv preprint arXiv:1807.04457*, 2018.

Data Mining Critical Infrastructure Systems: Models and Tools

Anika Tabassum, Supriya Chinthavali, Liangzhe Chen and B. Aditya Prakash

Abstract—**Critical infrastructures (CIs) such as power, water, transportation, and telecommunication are highly complex interacting systems that are vital to national security, economy and public life. They play an important role in several core urban computing challenges. Advances in computing resources and techniques have led to enormous progress in developing intelligent frameworks for analyzing these large heterogeneous systems. In this article, we survey state-of-the-art and foundational work in this upcoming area from a data mining perspective. We discuss basic concepts of CIs, their properties, impacts on them due to natural or human-caused disturbances and different computational methodologies used for modeling and understanding their behavior. We also discuss recent work that specifically deals with two critical sectors of CIs, namely power and transportation systems. Finally, we also describe several existing tools and methods that are used to facilitate decision making for domain operators, enable efficient and faster disaster response for federal and state agencies and help improve the security and resiliency of these CIs.**

Index Terms—**Critical Infrastructures, Urban Computing, Data Mining**

I. Introduction

URBAN computing is a process of measuring, modeling, analyzing and integrating complex heterogeneous data gathered from urban spaces [1]. The rapid growth of urbanization and modernization of people's lives have led to serious sustainability issues and hence there is a critical need for designing efficient and environment friendly systems, utilizing traffic flow in the city, situational awareness during extreme events to improve resiliency, public health, air pollution, etc. These complex and dynamic requirements give rise to several computational challenges in transportation, environment, cyber physical systems and internet of things [2], [3]. Some of these challenges include predicting power consumption [4], measuring air quality [5] and predicting, utilizing and controlling crowd/traffic flow in a city [6]–[8]. With the advent of social media, localizing and visualizing disaster events using such data has also proven to be a fertile ground for computational problems. For example, Sakaki et al. [9] proposed a classifier to monitor and detect earthquakes from Twitter and Yang et al. [10] designed a visualization of a four phase model using Twitter data (that represents content (what), location (where), time(when) and the user network (who) responding to the disaster). Public health issues like syndromic surveillance of a disease like influenza [11], and then controling it [12] are important problems as well.

Critical Infrastructure Systems: Critical infrastructure refers to systems, facilities, technologies and networks that are vital to security, public health and socio-economic well being of people. Clearly, they play an important role in many urban computing challenges. For example communication networks are inherently crucial for disaster response and controlling traffic flow. Hence, strengthening and maintaining secure and resilient Critical Infrastructure Systems (CISs) is a primary US national goal (even addressed through a presidential policy directive (PPD-21) [13]), and it requires proactive and coordinated efforts among federal, state, local, public and private owners and operators of CISs. Critical infrastructure networks (CIs) such as power, water, transportation, etc. are highly interdependent, and failure of one has a cascading effect on another which affect national security, economy and public health. Infact, a single vulnerable network can have a huge impact due to interdependencies. For example, the well-known 2003 Northeast blackout in U.S. [14] impacted multiple CIs. The massive power outage affects water and waste systems, transportation, communication, financial services, which cascaded to impact public health and food industries. Nearly 50 million people were affected and cause a loss of $5 billion to U.S. national economy [15]. Smart cities are extensively leveraging telecommunication technologies and cyber physical systems to provide a safe and a sustainable environment for increasing urban populations [16]. This also leads to an increasing risk of triggering cascading failures due to complex interactions and inter dependencies leading to debilitating impacts and creating urban computing challenges of cyber security, real-time situational awareness, handling traffic flow, meeting electricity demand, managing public health etc.

Data mining challenges: Modeling and analyzing such CISs gives rise to a rich and fruitful space of data mining challenges.

1) *Complexity.* CIs networks have a complex structure. As mentioned above, even a single CI network, e.g., electrical grid system consists of many underlying subsystems, *i)* power generator generates power using different types of fuel, *ii)* transmission network transfer power to different distribution substations over long distances, *iii)* distribution substations transfer power to local facilities and residential areas over the distribution grid, *iv)*Oil and Natural gas (NG) pipeline networks carry fuel to power generation stations. These pipeline networks consist of natural gas compressors,gas processing plants, NG terminals and other subsystems. These complex subsystems consist of large-scale data which is very useful for

Anika Tabassum and B. Aditya Prakash are with Department of Computer Science, Virginia Tech, e-mail: ({anikat1,badityap}@cs.vt.edu). Supriya Chinthavali is with Oak Ridge National Laboratory, e-mail: (chithavalis@ornl.gov). Liangzhe Chen is with Pinterest, e-mail: (liangzhechen@pinterest.com).

analyzing CIs.

2) *Heterogeneity.* CI networks are also extremely heterogeneous. They consist of many interdependencies like *i)* physical, where one infrastructure is physically connected or interdependent on another infrastructure e.g. power lines connected to water pumping stations, *ii)* geographical, where changes caused by local environmental events impact all CI components that are co-located, *iii)* cyber, state of infrastructure depends on information transmitted through information infrastructure, e.g., electronic and informational linkages. This heterogeneity and dependencies give rise to different types of nodes, edges, links, and multiple sources of information in the network [17].
3) *Dynamics.* CISs are also highly dynamic. Multiple incidents can cause failure of CI networks, e.g., loss of power, natural or human-made disaster which affect network in different states of operation varying with time. This dynamic property makes the system modeling more challenging.
4) *Scale.* For all these situations, designing scalable algorithms is a fundamental goal. These systems are in large scale and hence naturally give rise to 'big-data' problems.

Overview: In this article, we present the state-of-the-art research on models and tools used within CISs. This area of research is highly interdisciplinary, with connections to high-impact areas, like public safety/security, national economy, physics and power engineering and social media. We will first discuss various approaches to CI modeling. Then we mainly focus on two specific CI networks: power and transportation systems. We chose these networks since they are the heart of the sixteen CIs defined by the Department of Homeland Security (DHS) which have the potential to impact every other CIs (as discussed before, see the 2003 blackout example). In addition, these two areas have also seen a spate of recent work from a data mining perspective. We finally look into the methods that help improve situational awareness during disasters and then present some existing CI tools designed for helping domain experts in decision making and used by agencies such as national labs.

II. Modeling

CIs are highly interdependent and complex — failure of a single CI network can severely affect other CIs. Hence, in order to understand critical infrastructures to identify vulnerabilities, to protect them against threats and to support decision making, modeling and simulating them are essential. Modeling and simulating the interdependencies of CIs can be categorized into system dynamics-based, agent-based, network-based approaches [15].

System dynamics based approaches typically model CIs utilizing a *top-down* method to manage and analyze complex interdependencies based on domain knowledge of the particular system [15]. The main philosophy of such approaches is that for modeling a system, it is necessary to understand the behavior of a system. Different system-dynamics models have been proposed different sectors of CIs like telecommunication, petroleum, natural gas, and electric supply systems [18]–[20]. As an example, different models for electric grids have been proposed which represent system behavior using physical power equations and flow of electricity [20]–[22]. These models exhibit high-fidelity, and due to their complexity and highly detailed structure, they are also computationally expensive.

In contrast, agent-based modeling adopts a *bottom-up* approach which considers complex system behavior arising from the interaction of individual autonomous agents. For example, several agent-based systems [22]–[24] model power systems by considering the interaction and impacts of electric power markets on interdependencies of CIs. Similarly, to control and simulate traffic systems, different agent-based models have been proposed from a civil engineering perspective which consider environmental factors and vehicle crashes [25]–[27]. Balmer et al. [28] generate strategies for a traffic model by simulating the movement of agents, avoiding obstacles and generating congestion.

Finally, network-based approaches view CIs as *networks* where nodes represent different CIs components, and the links represent connections or edges among them. This approach requires less domain knowledge for modeling than the former, and hence they can be generalized to different systems. As a result while they are not high-fidelity and can not model all behaviors of the systems, they are great at modeling specific aggregate aspects. Hence they are also closer to data mining methodologies and problems. For instance, to ensure reliable broadcast in communication networks Duan et al. [29] studied interdependencies between communication and power grid networks and proposed an algorithm to handle both crash failures in communication and cascading failures in the power grid. As another example, Lee et al. created a system to generate CI networks [30] and model a cascade of failures among different CIs based on their physical topology and temporal dependencies. Chen et al. [31] proposed an optimization algorithm named *FASCINATE* to infer cross-layer dependencies in multilayered CI networks where each layer consists of a different CI network. To infer the cross-layer relation between two different networks, they viewed multilayered connections as a collective collaborative filtering problem. The overall approach is shown in Fig 1.

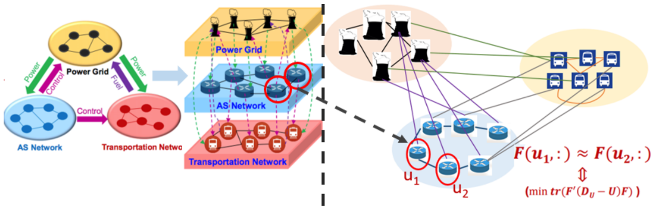


Fig. 1. An example of a network-based approach to system modeling. The goal of *FASCINATE* proposed by Chen et al. [31] is to infer dependecies among the CI networks. Left figure shows how a CI network is converted into a cross-layered network. Right figure shows how they find out the hidden dependencies across the transportation and power grid networks by viewing it as a collective collaborative filtering problem.

III. Example CI: Power Systems

In this section, we provide a few examples of work on CI systems dealing with power and energy, from a network modeling perspective. As representative work, we focus on methods that identify vulnerable facilities to protect against unknown natural disasters (non-adversarial) and to protect the system against adversarial attacks with known patterns and strategies.

A. Identifying vulnerable facilities

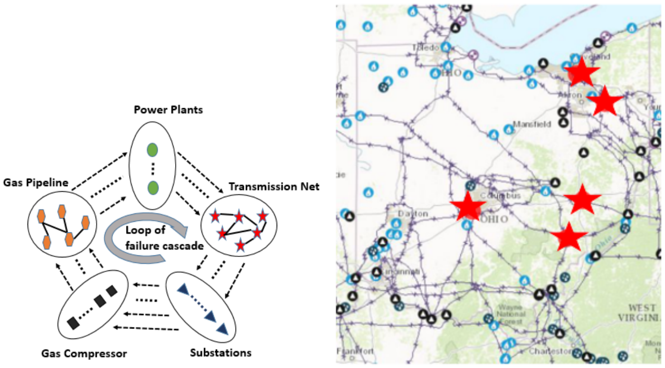
Identifying vulnerable critical facilities in a power system is necessary to protect and enhance them against unknown natural disasters. The state-of-the-art can be divided into mainly two different techniques: using only the network structure and/or incorporating failure cascade dynamics as well.

Several algorithms have been proposed in order to identify critical nodes using only the network structure [32]–[34]. Arianos et al. [32] introduced the concept of using geodesic distance for power flow to calculate resiliency of power grids. Chen et al. [33] proposed an algorithm *OPERA* for the connectivity control problem. It aims to find a set of optimal nodes that maximizes the impact on an interdependent CI network consisting of physical, control and communication layers. In addition, they developed *SUBLINE*, to unify a family of prevalent subgraph connectivity measures to quantify network dependency of the graph and subsequently use it for identifying the optimal nodes.

Additionally, incorporating failure dynamics can help prevent catastrophic failures of the whole system. The 2003 blackout shows an instance (see Section I for detail) where a transmission line failure cascaded to failures of water, waste-treatment, and communication systems. Buldyrev et al. [34] developed a framework based on mutually connected clusters to study cascading failures in interdependent networks. They analyzed the presence of a giant connected cluster under simple random failures on Erdos-Renyi networks and show a phase transition and a critical threshold. In contrast, Chen et al. [35] developed an algorithm *HOTSPOTS* to model more complicated failure cascades and identify critical nodes that may lead to substantial failures. Their heterogeneous network consists of power plants, substations, transmissions and gas compressors (see Fig. 2). First, they propose a 'path-based' failure cascade model on this complex system representing how every component of a CI network interacts with each other. Instead of the typical neighbor-based cascade models they propose a novel path-based failure cascade (*F-CAS*). Second, given the *F-CAS* model, they formulate an optimization algorithm to identify a set of critical transmission nodes whose failure will maximize the number of failed substations (another CI network). Finally, they propose a dominator-tree based approach to solve this problem efficiently.

B. Protecting power system against attacks

Next we discuss several works that studied how to detect an attack, and protect and/or to reduce the effect of an adversarial attack on power grid. To detect an attack or a node failure in the network, Hooi et al. [36] proposed an online anomaly detection algorithm *GRIDWATCH* that can help a sensor to detect a failure in the electrical grid. Using *GRIDWATCH* they also suggested an optimization which can maximize the probability of detecting a failure in power grid within a given budget.



(a) Heterogeneous network (b) Critical transmission nodes

Fig. 2. An example of identifying vulnerable facilities in power system using both network structure and failure dynamics (Section III). Fig. (a) is an overview of *HOTSPOTS* proposed by Chen et al. [35] which shows a heterogeneous network in a power system which consists of power plants, transmission networks, substations, gas compressor, and gas pipelines. Fig. (b) shows the maximum failed transmission nodes in OH that the algorithm identifies and that these nodes are close to the failed nodes during 2003 blackout [14].

To enhance grid resiliency it is necessary to understand which nodes should be protected first under targeted attacks. Using the giant connected cluster formulation of Buldyrev et al. [34] discussed above, Huang et al. [37] developed a framework to understand robustness in interdependent networks under degree-based targeted attacks. Their main idea was to map a targeted network to a random one (on a different network). Their findings show that protecting the higher degree nodes with low probability to fail can significantly improve robustness of interdependent networks.

Finally, if an attack happens it is necessary to reduce its impact on the interdependent network. Strategies for this task have been proposed by many papers [38], [39]. Wang et al. [38] proposed a load redistribution approach where a failure at node i redistributes its load to its neighboring nodes to reduce the impact of an attack. Ouyang et al. [39] proposed a tri-level optimization problem which maximizes the resiliency of the system and also minimize loss. The inner level optimizes the damaged components to repair, middle level identifies the most disruptive attack, and outer level optimizes the defense decision.

IV. Example CI: Transportation Systems

In transportation systems, research has been done on predicting traffic on roads, traffic states like accidents, road constructions as well as several approaches to improve congestion control.

A. Traffic flow and state prediction

Traffic flow is the study of interactions of vehicles with the traffic infrastructure, such as traffic control devices, highways and traffic signs. Predicting traffic flow and states can help improve an intelligent transportation system and prevent congestion. In order to predict traffic flow on roads, it is necessary to understand the influence of road segments based on propagating congestion. Anwar et al. [40] developed an algorithm to identify the most congested areas on traffic using the road intersection network, the number of vehicles passing during green lights and the ratio of effective usage of green light time on each road segment. Many different algorithms have also been proposed in order to forecast traffic flow at time $t+k$ from the recent traffic flow data up to time t [41]–[43]. Wu et al. [41] suggest a random effect model integrating the temporal factors while Moretti et al. [42] developed a hybrid ensemble technique using an artificial neural network along with a statistical regression approach. As another example, Zheng et al. [43] predicted traffic flow using occupancy data from buildings, instead of using traffic data directly. Traffic states govern traffic flow and can be categorized into 1) traffic infrastructure states (TIS), e.g., weather, presence of accidents, roadworks and 2) traffic flow state (TFS), e.g., flow rate, density, speed. From the partial observations of TFS and TIS at several periods of different traffic links Gu et al. [45] designed a model to estimate the traffic states using expectation maximization and kalman filters. Instead, Adhikari et al. [44] formulate the network state detection problem as an inference problem given partial state data (from a crowd-sourced app like Waze). They develop a near-optimal efficient algorithm, *GRAPHSTATEINF*, which tries to find a set of failed nodes in a network using observed failed nodes and correlations among failure nodes in the network. Their main idea was to leverage the information-theoretic MDL (minimum description length) principle, which searches for the 'best' set of failures which 'explains' the given partial set with the minimum encoding cost. This approach is useful in predicting the impact of networks due to congestion (shown in Fig. 3).

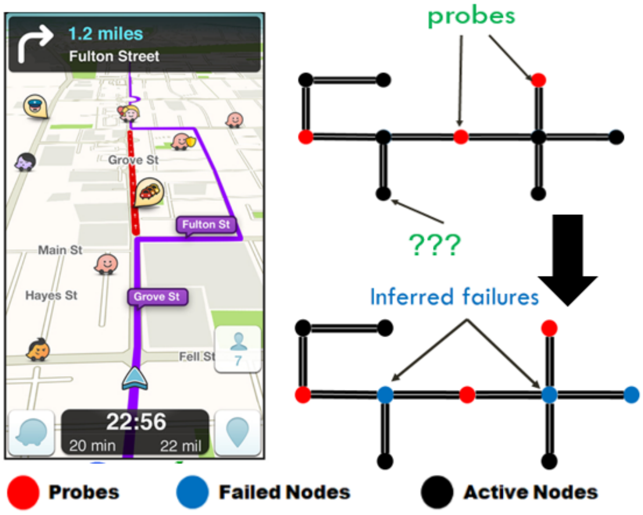


Fig. 3. Traffic state inference using partial data. *GRAPHSTATEINF* proposed by Adhikari et al. [44] dynamically identifies all the failed nodes in the road network by leveraging a subset observations of failed nodes. The left figure shows an example of a crowd-sourced app using which the user reports incidents (the so-called probes of failed nodes) (left figure). In the right figure, a toy road network showing the observed partially failed nodes (marked as red) and inferred unobserved failed nodes (marked as blue).

B. Congestion tracking and control

To build an improved traffic system it is necessary to control or reduce traffic congestion. For this, detecting any congestion is the first step. Anwar et al. [46] model congestion on road networks as partitions of these networks such that road blocks in each partition are homogeneous and have similar congestion (see Fig. 4, where roads in the same round block have similar congestion). However, traffic congestion varies with time and re-partitioning them at each time-step is computationally expensive. Hence they also develop an algorithm to incrementally update the congested partitions at a new time point based on previous time and current traffic data. Their main idea is to find the unstable nodes and assign these nodes to a block which maximizes the number of bounded cycles in a block.

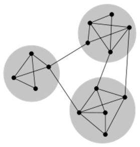


Fig. 4. Congestion Example

In order to control congestion, Sundar et al. [47] designed an automatic traffic signal control system from the traffic density in the route. They used signals from RFID sensors inserted in every vehicle, which transmits messages to the reader. Along with congestion control, this system is also able to detect stolen vehicles and clear traffic signal automatically for the emergency vehicles.

V. Facilitating Decision Making

Various CI systems and tools have been developed to aid the domain experts in taking multifaceted decisions for emergency management, improve situation awareness as well as for budget planning purposes. Next we give an overview of this space.

A. Improving situation awareness

The idea of situation awareness is to predict emergency and differentiate casual events from non-casual events, and many approaches use crowdsourcing in some way for this along with other techniques. Liang et al. [48] built a classifier to distinguish flooded areas from non-flooded areas from satellite images. The authors developed a semi-supervised learning algorithm which divides the satellite image into several patches based on the proximity and intensity of the pixels. A user is asked to label a few patches and based on that the classifier automatically classifies all other patches. Fig 5 shows the steps of the algorithm. They used crowd-sourced knowledge for getting user feedback for their classifier instead of using domain expert to label the image patches.

Similarly, in the context of traffic data, Hooi et al. [49] proposed an algorithm to find out traffic accidents by detecting change points from sensor data. Huang et al. [50] designed a crowd-sourcing based anomaly prediction system which allows a user to report urban anomalies they encountered. Based on these historical anomaly data they developed a Bayesian inference model to understand dependency among regions regarding the anomaly distribution. Next, they built a Markov model to learn state transitions between normal and

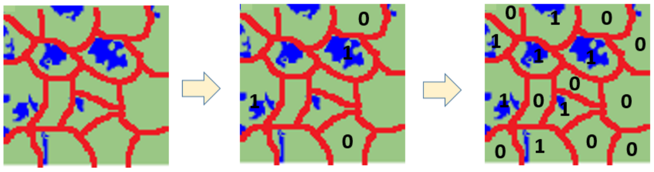


Fig. 5. An example of human-guided flood detection technique from satellite images proposed by Liang et al. [48] for improving situation awareness (see Sec V). In Step 1 the algorithm divides their images into several patches. In Step 2, some of the patches of the image are labeled using crowdsourced feedback. In Step 3, based on the user feedback, the algorithm labels other patches as flood or non-flooded areas.

anomaly data and predict the state of the next time slot. Muralidhar et al. [51] also developed an online monitoring system *ILLIAD* for anomaly detection and state estimation in cyber physical systems (CPS), e.g., wireless and wired networks. They combined model-based and data driven approaches to learn invariant functional relationships between components of the CPS (shown to represent the underlying network structure of the CPS). Next they checked for the violation of any of these invariant relationships over time which was then treated as an anomaly (see Fig. 6).

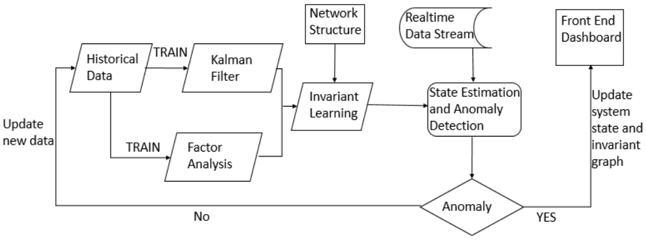


Fig. 6. An example to improve situation awareness in CPS (wireless and wired networks). The framework of *ILLIAD* proposed by Muralidhar et al. [51]. They used Kalman filters (model based) and autoregression and latent factor based (data-driven) methods to learn the invariant functional relationship between the components and used that for state estimation and anomaly detection.

Social media data can also be leveraged to detect disaster-prone areas. Mcclendon et al. [52] show how social media data can support the decision for emergency management by categorizing disaster-affected areas. Using Twitter users as sensors Sakaki et al. [9] designed a classifier to detect target events and a probabilistic spatiotemporal model to find the center and trajectory of the event. Zhao et al. [53] proposed a model to predict spatial events in social media considering different spatial locations and various spatial relationship with the task. Farag et al. [54] developed an event model that can automatically capture the event information and incorporated the model into a focused crawler algorithm which can identify the web pages relevant to that event. To detect the disaster prone-areas and for fast emergency response using social media, it is also necessary to identify trustworthy users whose contents are influential. Vedula et al. [55] developed an unsupervised algorithm to identify the trustworthy and influential users from the network during a crisis.

B. Other CIS systems and tools

Several CIs tools have been developed to support the decision-making process. Argonne National Lab has developed a risk-based decision support system and a simulation model named Critical Infrastructure Protection and Decision Support System (CIPDSS) [56] to protect CI systems against vulnerabilities, natural or human-made disasters, etc. For studying energy development and impacts of climate Los Alamos National Lab (LANL) designed Climate-Energy Assessment for Resiliency (*CLEAR*) model which enables to assess the interdependency of CIs regarding their relationship with climate. Figure 7 shows an example of *CLEAR* [57] assessing CO_2 emission in transportation and energy sectors. In addition, a toolkit *URBAN-NET* developed by Lee et al. [30] integrates network construction, visualization, failure cascade modeling, and a simulator to identify critical facilities. They generated the physical CI networks from disparate data sources which contains location and information of CI components. For modeling and simulation, they considered the system into three different categories: topology-based, simulation-based and monitoring based analysis. In the topology-based analysis, they consider only physical CI interdependencies and compute the importance of each node and link based on their interdependencies. In simulation-based analysis, they also incorporated temporal dependencies such as the restoration period of network failure, and capacity of a CI component to handle the failure of its interdependent network. They showed an example with a road-gas network where each node importance is based on its efficiency of transportation as well as how well it is reachable to gas stations. Finally they also created a visualization to show the critical nodes and edges (shown in Fig. 8).

There are other such tools have been developed by Oak Ridge National Lab (ORNL) to facilitate decision making for urban infrastructures. *URBAN-CAT* [58] has been developed to understand the impacts of climate change, *URBAN-MET* [59] has been designed to study interactions between urban and environmental systems, *LANDSCAN* [60] has been developed to model the distribution of population and settlement based on demographic and remote sensing imagery data. Besides, several CIs datasets are also available to aid research and decision making: a geospatial and US domestic infrastructure HSIP gold data [61], NHDPlus and USGS hydrology data [62], [63], and EIA energy data [64].

VI. Conclusion

In this survey, we presented an overview of state-of-the-art in CI research from an urban computing and data mining perspective. First, we describe the state-of-the-art approaches that have been used in CIs for modeling dynamics in the system. First we discussed some of the different frameworks for modeling CI systems in general. Second, we discuss the data mining problems related to two vital sectors in CIs, electric grid systems, and transportation systems. Within the power grid systems, we showcase existing techniques to identify vulnerable facilities and to protect the system from adversarial attacks. Within the transportation domain, we

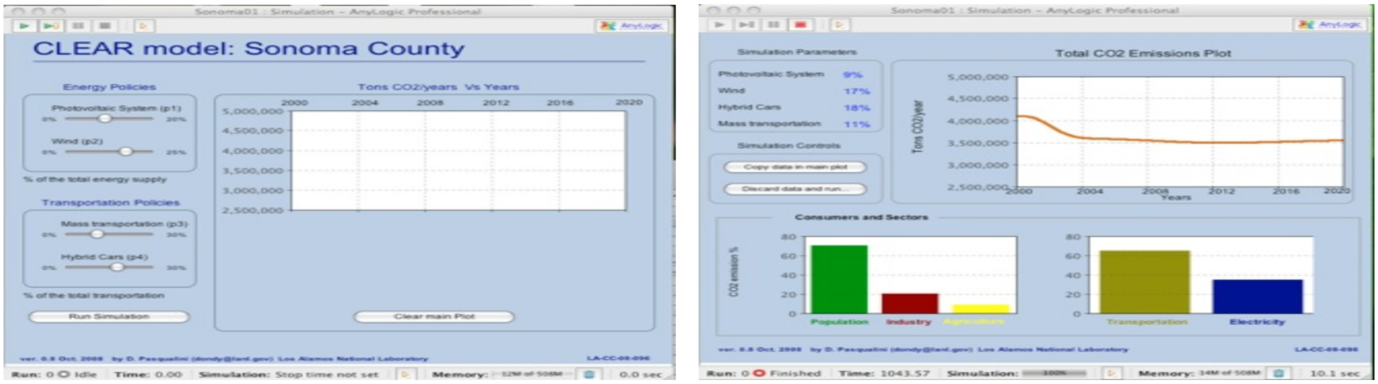
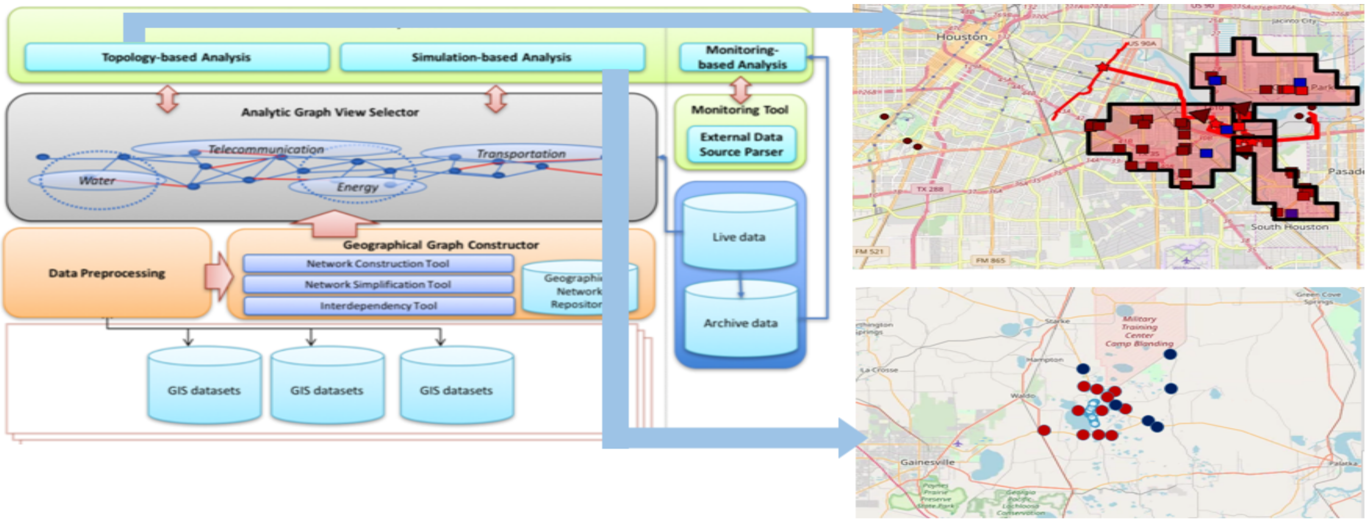


Fig. 7. A screenshot of the interface of *CLEAR* designed by LANL [57] showing emission of CO_2 in typical sectors of CIS. In the left figure a user can choose the energy and transportation policy, whereas the right figure shows the simulation result, i.e., the amount of CO_2 emission in different sectors based on the chosen policy.


Topology-based Analysis
Simulation-based Analysis
Monitoring-based Analysis
Analytic Graph View Selector
Telecommunication
Transportation
Water
Energy
Monitoring Tool
External Data Source Parser
Live data
Archive data
Data Preprocessing
Geographical Graph Constructor
Network Construction Tool
Network Simplification Tool
Interdependency Tool
Geographical Network Repository
GIS datasets
GIS datasets
GIS datasets
Houston
Pasadena
South Houston
Gainesville
Military Training Center Camp Blanding

s

Fig. 8. Overview of *URBAN-NET* by Lee et al. [30]. Left fig. shows how *URBAN-NET* is collecting GIS data and preprocessing then to analyze CI networks. Top right fig. shows a topology-based example where the system identifies the vulnerable CI networks which fails due to the failure of some initial seed nodes. Bottom right fig. shows a simulation-based analysis where it identifies the critical nodes which fail after a certain time due to interdependency.

presented algorithms for traffic states and flow prediction and to control congestion. Finally, we also described some popular tools and techniques, that have been developed to ease the decision-making process for domain experts.

There are several open problems and this is a rich domain with high potential for interdisciplinary impact. For example, in context of energy systems, some of these problems include: *1)* Interpreting or explaining the behavior of CIs models, for e.g., are the algorithms able to identify critical facilities from the system; *2)* In terms of modeling, federal entities such as Department of Energy (DoE) focus on improving understanding of how extreme events (such as hurricanes and wildfires) impact the production of electricity and power equipment (such as flooded substations, downpoles etc.) [65]. Since renewable generation such as solar and wind tends to be highly intermittent, there is a lot of interest to resolve this issue to aim towards uninterrupted electricity supply and improve sustainability [66]; *3)* Impacts of electricity production caused by outages in gas pipe lines is also being heavily pursued due to growth in natural gas fuel production within the US; *4)* Another area of investigation is to understand the impacts of power restoration due to disruptions in the transportation infrastructure.

Acknowledgements: This article is based on work partially supported by the NSF (IIS-1353346, CAREER IIS-1750407), the NEH (HG-229283-15), ORNL, the Maryland Procurement Office (H98230-14-C-0127), and a Facebook faculty gift.

References

[1] "Urban dynamics institute." [Online]. Available: https://udi.ornl.gov/
[2] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 3, p. 38, 2014.
[3] F. Salim and U. Haque, "Urban computing in the wild: A survey on large scale participation and citizen engagement with ubiquitous computing, cyber physical systems, and internet of things," *International Journal of Human-Computer Studies*, vol. 81, pp. 31–48, 2015.
[4] Y. Weng, R. Negi, C. Faloutsos, and M. D. Ilić, "Robust data-driven state estimation for smart grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1956–1967, 2017.
[5] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1436–1444.

[6] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction." in *AAAI*, 2017, pp. 1655–1661.

[7] Y.-J. Wu, F. Chen, C. Lu, B. Smith, and Y. Chen, "Traffic flow prediction for urban network using spatio-temporal random effects model," in *91st Annual Meeting of the Transportation Research Board (TRB)*, 2012.

[8] D. Liu, D. Weng, Y. Li, J. Bao, Y. Zheng, H. Qu, and Y. Wu, "Smartadp: Visual analytics of large-scale taxi trajectories for selecting billboard locations," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 1–10, 2017.

[9] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 851–860.

[10] S. Yang, H. Chung, X. Lin, S. Lee, L. Chen, A. Wood, A. L. Kavanaugh, S. D. Sheetz, D. J. Shoemaker, and E. A. Fox, "Phasevis1: What, when, where, and who in visualizing the four phases of emergency management through the lens of social media." in *ISCRAM*, 2013.

[11] L. Chen, K. T. Hossain, P. Butler, N. Ramakrishnan, and B. A. Prakash, "Syndromic surveillance of flu on twitter using weakly supervised temporal topic models," *Data mining and knowledge discovery*, vol. 30, no. 3, pp. 681–710, 2016.

[12] Y. Zhang, A. Ramanathan, A. Vullikanti, L. Pullum, and B. A. Prakash, "Data-driven immunization," in *Data Mining (ICDM), 2017 IEEE International Conference on*. IEEE, 2017, pp. 615–624.

[13] "Presidential policy directive – critical infrastructure security and resilience." [Online]. Available: https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil

[14] B. Liscouski and W. Elliot, "Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations," p. 86, 2004.

[15] M. Ouyang, "Review on modeling and simulation of interdependent critical infrastructure systems," *Reliability engineering & System safety*, vol. 121, pp. 43–60, 2014.

[16] H. Boyes, R. Isbell, and T. Watson, "Critical infrastructure in the future city," in *International Conference on Critical Information Infrastructures Security*. Springer, 2014, pp. 13–23.

[17] P. Pederson, D. Dudenhoeffer, S. Hartley, and M. Permann, "Critical infrastructure interdependency modeling: a survey of us and international research," *Idaho National Laboratory*, vol. 25, p. 27, 2006.

[18] G. P. O'Reilly, A. Jrad, A. Kelic, and R. LeClaire, "Telecom critical infrastructure simulations: Discrete-event simulation vs. dynamic simulation how do they compare?" in *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*. IEEE, 2007, pp. 2597–2601.

[19] T. Brown, "Multiple modeling approaches and insights for critical infrastructure protection," *NATO Security through Science Series D-Information and Communication Security*, vol. 13, p. 23, 2007.

[20] X. Xu, H. Jia, H.-D. Chiang, D. Yu, and D. Wang, "Dynamic modeling and interaction of hybrid natural gas and electricity supply system in microgrid," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1212–1221, 2015.

[21] S.-K. Kim, J.-H. Jeon, C.-H. Cho, J.-B. Ahn, and S.-H. Kwon, "Dynamic modeling and control of a grid-connected hybrid generation system with versatile power transfer," *IEEE transactions on industrial electronics*, vol. 55, no. 4, pp. 1677–1688, 2008.

[22] H. A. Song, B. Hooi, M. Jereminov, A. Pandey, L. Pileggi, and C. Faloutsos, "Powercast: mining and forecasting power grid sequences," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 606–621.

[23] D. C. Barton, E. D. Eidson, D. A. Schoenwald, K. L. Stamber, and R. K. Reinert, "Aspen-ee: an agent-based model of infrastructure interdependency," *SAND2000-2925. Albuquerque, NM: Sandia National Laboratories*, 2000.

[24] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury, "Epochs: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 548–558, 2006.

[25] H. A. Aziz and S. V. Ukkusuri, "Integration of environmental objectives in a system optimal dynamic traffic assignment model," *Computer-Aided Civil and Infrastructure Engineering*, vol. 27, no. 7, pp. 494–511, 2012.

[26] H. A. Aziz, S. V. Ukkusuri, and S. Hasan, "Exploring the determinants of pedestrian–vehicle crash severity in new york city," *Accident Analysis & Prevention*, vol. 50, pp. 1298–1309, 2013.

[27] F. Zhu, H. A. Aziz, X. Qian, and S. V. Ukkusuri, "A junction-tree based learning algorithm to optimize network wide traffic control: A coordinated multi-agent framework," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 487–501, 2015.

[28] M. Balmer, N. Cetin, K. Nagel, and B. Raney, "Towards truly agent-based traffic and mobility simulations," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*. IEEE Computer Society, 2004, pp. 60–67.

[29] S. Duan, S. Lee, S. Chinthavali, and M. Shankar, "Best effort broadcast under cascading failures in interdependent critical infrastructure networks," *Pervasive and Mobile Computing*, vol. 43, pp. 114–130, 2018.

[30] S. Lee, L. Chen, S. Duan, S. Chinthavali, M. Shankar, and B. A. Prakash, "Urban-net: A network-based infrastructure monitoring and analysis system for emergency management and public safety," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2600–2609.

[31] C. Chen, H. Tong, L. Xie, L. Ying, and Q. He, "Fascinate: Fast cross-layer dependency inference on multi-layered networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 765–774.

[32] S. Arianos, E. Bompard, A. Carbone, and F. Xue, "Power grid vulnerability: A complex network approach," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 1, p. 013119, 2009.

[33] C. Chen, J. He, N. Bliss, and H. Tong, "On the connectivity of multi-layered networks: Models, measures and optimal control," in *2015 IEEE International Conference on Data Mining*. IEEE, 2015, pp. 715–720.

[34] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, p. 1025, 2010.

[35] L. Chen, X. Xu, S. Lee, S. Duan, A. G. Tarditi, S. Chinthavali, and B. A. Prakash, "Hotspots: Failure cascades on heterogeneous critical infrastructure networks," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 1599–1607.

[36] B. Hooi, D. Eswaran, H. A. Song, A. Pandey, M. Jereminov, L. Pileggi, and C. Faloutsos, "Gridwatch: Sensor placement and anomaly detection in the electrical grid," 2018.

[37] X. Huang, J. Gao, S. V. Buldyrev, S. Havlin, and H. E. Stanley, "Robustness of interdependent networks under targeted attack," *Physical Review E*, vol. 83, no. 6, p. 065101, 2011.

[38] J.-W. Wang and L.-L. Rong, "Cascade-based attack vulnerability on the us power grid," *Safety Science*, vol. 47, no. 10, pp. 1332–1336, 2009.

[39] M. Ouyang and Y. Fang, "A mathematical framework to optimize critical infrastructure resilience against intentional attacks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 11, pp. 909–929, 2017.

[40] T. Anwar, C. Liu, H. L. Vu, and M. S. Islam, "Roadrank: Traffic diffusion and influence estimation in dynamic urban road networks," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 1671–1674.

[41] Y.-J. Wu, F. Chen, C.-T. Lu, and S. Yang, "Urban traffic flow prediction using a spatio-temporal random effects model," *Journal of Intelligent Transportation Systems*, vol. 20, no. 3, pp. 282–293, 2016.

[42] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, 2015.

[43] Z. Zheng, D. Wang, J. Pei, Y. Yuan, C. Fan, and F. Xiao, "Urban traffic prediction through the second use of inexpensive big data from buildings," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 1363–1372.

[44] B. Adhikari, P. Rangudu, B. A. Prakash, and A. Vullikant, "Near-optimal mapping of network states using probes," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 108–116.

[45] Y. Gu, Z. Qian, and G. Zhang, "Traffic state estimation for urban road networks using a link queue model," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2623, pp. 29–39, 2017.

[46] T. Anwar, C. Liu, H. L. Vu, and M. S. Islam, "Tracking the evolution of congestion in dynamic urban road networks," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 2323–2328.

[47] R. Sundar, S. Hebbar, and V. Golla, "Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1109–1113, 2015.

[48] J. Liang, P. Jacobs, and S. Parthasarathy, "Human-guided flood mapping: From experts to the crowd," in *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 2018, pp. 291–298.

[49] B. Hooi, L. Akoglu, D. Eswaran, A. Pandey, M. Jereminov, L. Pileggi, and C. Faloutsos, "Changedar: Online localized change detection for sensor data on a graph," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 507–516.

[50] C. Huang, X. Wu, and D. Wang, "Crowdsourcing-based urban anomaly prediction system for smart cities," in *Proceedings of the 25th ACM international on conference on information and knowledge management*. ACM, 2016, pp. 1969–1972.

[51] N. Muralidhar, C. Wang, N. Self, M. Momtazpour, K. Nakayama, R. Sharma, and N. Ramakrishnan, "illiad: Intelligent invariant and anomaly detection in cyber-physical systems," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 3, p. 35, 2018.

[52] S. McClendon and A. C. Robinson, "Leveraging geospatially-oriented social media communications in disaster response," *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)*, vol. 5, no. 1, pp. 22–40, 2013.

[53] L. Zhao, F. Chen, C.-T. Lu, and N. Ramakrishnan, "Multi-resolution spatial event forecasting in social media," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 689–698.

[54] M. M. Farag, S. Lee, and E. A. Fox, "Focused crawler for events," *International Journal on Digital Libraries*, vol. 19, no. 1, pp. 3–19, 2018.

[55] N. Vedula, S. Parthasarathy, and V. L. Shalin, "Predicting trust relations within a social network: A case study on emergency response," in *Proceedings of the 2017 ACM on Web Science Conference*. ACM, 2017, pp. 53–62.

[56] "Critical infrastructure protection decision support system (cipdss)." [Online]. Available: http://www.bwbush.io/projects/cipdss.html

[57] "Climate-energy assessment for resiliency (clear)." [Online]. Available: https://public.lanl.gov/dp/CLEAR.html

[58] "Urban climate adaptation tool (urban-cat)." [Online]. Available: https://udi.ornl.gov/content/urban-climate-adaptation-tool-urban-cat

[59] "Urban-met: Modeling urban energy savings scenarios using earth system microclimate and urban morphology." [Online]. Available: https://studylib.net/doc/11896703/urban-met--modeling-urban-energy-savings-scenarios-using

[60] "Landscan." [Online]. Available: https://landscan.ornl.gov/

[61] "Hsip gold dataset." [Online]. Available: https://gii.dhs.gov/HIFLD/hsip-guest

[62] "Nhdplus." [Online]. Available: http://www.horizon-systems.com/nhdplus/

[63] "Usgs water data." [Online]. Available: https://waterdata.usgs.gov/nwis/rt

[64] "Us energy information administration." [Online]. Available: https://www.eia.gov/

[65] E. S. Blake, T. B. Kimberlain, R. J. Berg, J. P. Cangialosi, and J. L. Beven Ii, "Tropical cyclone report: Hurricane sandy," *National Hurricane Center*, vol. 12, pp. 1–10, 2013.

[66] "Us department of energy (doe) advanced modeling grid research program." [Online]. Available: https://www.energy.gov/sites/prod/files/2018/07/f53/1.1.1%20NA%20Grid%20Resil%20Modeling%20Panel%20-%20Ghassemian%2C%20DOE.pdf

An Artificial Neural Network Approach for the Detection of Abnormal Heart Rhythms

Eyhab Al-Masri

***Abstract*— Automatically classifying arrhythmias remains a fundamental measure during the diagnosis or detection of cardiac abnormalities. In this paper, we propose a method to accurately classify ECG arrhythmias based on artificial neural networks (ANNs). In particular, we analyze three different heart rhythms including: (a) normal sinus rhythm, bradycardia and tachycardia. ECG Feature extraction and time-frequency analysis have proven to be useful in identifying arrhythmias. However, the accuracy in classifying ECG signals depends significantly on a large number of features that can be extracted. In this study, we attempt to optimize the number of features that are required to successfully classify ECG signals correctly. Throughout the paper, we present results from testing our backpropagation neural network (BPNN) with the existing MIT/BIH Arrhythmia database. The overall performance of our method demonstrates a success rate of 98.70% in identifying the correct type of cardiac arrhythmias.**

***Index Terms*— arrhythmia detection, arrhythmia classification, neural networks, backpropagation, heartbeats, non-stationary ECG signals, RR-intervals, MIT/BIH**

I. Introduction

IDENTIFYING correctly and accurately the type of an electrocardiogram (ECG) signal is an active research area in biomedical engineering and require significant knowledge in signal processing. Researchers have explored numerous complex algorithms for analyzing and identifying ECG signals including power spectrum analysis [1,2], principle component analysis [3], signal analysis [4], Hilbert transform analysis [5,6], continuous and discrete wavelet transforms, and adaptive filtering [7]. Other complex techniques involving feature extraction have also been applied such as template matching [8], markov models [9], neural networks [10-15], among many other recognition algorithms [16, 25-39].

Although there have been numerous research efforts in detecting and recognizing ECG wave abnormalities [17-24, 40-42], optimizing the feature set required for detecting heartbeat abnormalities without sacrificing the accuracy and success rates has often been ignored. Hence, it would be desirable to determine this optimal set such that it can be used by neural network algorithms for the detection of heartbeat abnormalities while maintaining high accuracy and classification rates.

In an effort to solve existing research problems and limitations with the current state-of-the-art, we introduce an Artificial Neural Network (ANN) solution that is capable of detecting ECG heartbeat abnormalities that can be used in real-time for ECG monitoring systems. A fundamental factor in achieving this goal is mainly dependent on finding the correct number and types of features (or parameters) that represent the various ECG signal conditions within an acceptable discrimination capability.

The rest of the paper is organized as follows. Section II presents an overview of the methodology applied in this research work. Section III presents the results we have obtained from applying our neural-network solution using the obtained dataset. We also discuss and provide insights on the use of ANN in the detection of heartbeat abnormalities in the same section. Finally, a conclusion and future work directions are provided in Section IV.

II. Methodology

The Massachusetts Institute of Technology (MIT) and Beth Israel Hospital (MIT/BIH) Arrhythmia Database [20] is one of a handful data sources that can be used by researchers and data scientists as test material for the evaluation of arrhythmia detection and classification. We used the MIT/BIH arrhythmia database's annotated records [20] for evaluating the developed neural network classifier. The dataset included data from forty eight patients with duration of thirty minutes recordings for each patient. MIT/BIH's data collection was completed using two leads: (a) modified limb lead II and (b) modified lead VI of surface ECG [20]. According to MIT/BIH, these two leads provide an adequate representation of complex QRS waveforms, artifacts and conduction abnormalities [20].

For the purpose of our study, we have randomly selected data representing thirty three patients. The random selection of such records was influenced by the fact that we would like to choose records that represent the three main types of arrhythmia that we would like to classify using our neural network and these include: (a) normal sinus rhythm, (b) bradycardia and (c) tachycardia ECG signals. All the recordings in MIT/BIH arrhythmia database were sampled at a frequency of 360Hz. In addition, the dataset contains an annotation file representing the "truth" identification provided by two or more expert cardiologists [20]. These "truth" annotations are the labels that we can then use for training, evaluating and testing our neural network classifier.

Our methodology for classifying arrhythmia involved two main phases: (a) a preprocessing phase and (b) a classification phase. The scope of this study focuses on particular set of

Eyhab Al-Masri is with School of Engineering and Technology, University of Washington Tacoma, Tacoma, WA 98402 USA. (e-mail: ealmasri@uw.edu).

heartbeat abnormality (i.e. arrhythmia) as predetermined by the dataset and identified using the truth labels. Table I outlines the standard components (i.e. features) including waves and intervals that can be identified in a normal electrocardiogram. Our main goal for using the BPNN classification model is to optimize the number of features (i.e. wave types in this case) while maintaining high accuracy in identifying the type of heartbeat (e.g. normal, tachycardia or bradycardia).

Our BPNN's preprocessing phase analyzes an ECG signal and detects information about the QRS complexes. This includes extracting information such as Q-wave, R-wave, T-wave, U-wave, PR interval, QRS duration, among others. Once these individual elements of an ECG signal are identified, the following task is to identify the RR-interval. An RR-interval, as described in Table 1, requires examination of two ECG signals at a time since the interval spans from the R-wave of the first ECG signal to the R-wave of another (preceding or following) ECG signal. In our case, we examined a following R-wave. That is, we examine the current ECG signal, extract the R-wave and then extract the next ECG signal information, determine the R-wave and then construct or calculate the RR-interval. This RR-interval provides valuable information and is used to that enhance the accuracy of the neural network classifier.

TABLE I
COMPONENTS OF AN ECG SIGNAL

Waves, Intervals & Segments	Description
P-wave	depolarisation of atria
Q-wave	(small) negative wave preceding R-wave
R-wave	depolarisation of ventricles
T-wave	repolarisation of ventricles
U-wave	(small) positive wave following T-wave
PR interval	start of P-wave to start of QRS complex
PR segment	end of P-wave to start of QRS complex
QRS duration	start to end of QRS complex
QT interval	start of Q wave to end of T-wave
RR interval	from an R-wave to next (consecutive) R-wave

The BPNN algorithm uses this information extracted (i.e. features) from an ECG signal to then classify whether an ECG signal is normal or abnormal. In specific, the BPNN attempts to identify the type of ECG signal or classify it as either a (a) sinus rhythm, (b) bradycardia or (c) tachycardia. The features extracted during preprocessing provide valuable details that help the ANN in classifying a given heartbeat. For instance, a typical normal heartbeat (i.e. normal sinus rhythm) is determined using a threshold that is based on the number of heart beats per minute. That is, if the heartbeat is greater than or equal to sixty beats per minute (bpm) and less than or equal to one hundred, an electrocardiogram signal is said to be normal (or a sinus rhythm). Furthermore, a heartbeat rate exceeding or higher than one hundred beats per minute is often referred to as tachycardia. In addition, a heartbeat rate that is lower than sixty is often referred to as bradycardia. It is assumed that the discussed heartbeat rates are calculated when an adult patient is at rest (e.g. not walking or running). This information is vital with respect to identifying the correct type of arrhythmia.

The rate at which a heart beats provides vital information whether there are abnormalities in the heart or not. In the case of a tachycardia, for example, a heartbeat rate is greater than one hundred beat per minute which translates into having shorter or small RR-intervals. On the other hand, in the case of a bradycardia, the RR-intervals will be longer than that of a normal ECG. The neural network uses this critical information to be able to determine whether a given ECG signal, for example, is classified as normal, bradycardia or tachycardia.

Classifying heartbeats into one of the three categories: normal sinus rhythm, bradycardia, or tachycardia is the chief objective of this research study. However, to perform this process properly, the neural network classifier needs to have high accuracy and produces truthful classification. Therefore, the input data supplied to the neural network should not contain contradictory information that might confuse the classifier yielding to inaccurate classifications. In addition, increasing the robustness of the classification rates of the neural network depends on the extraction and collection of multiple hidden features from the QRS complexes. The more features translates into more accurate findings. However, it is important that this set of features is optimized such that it does not overwhelm the neural network and reduces any overheads with respect to the time it take to process the input data.

In order to properly train and test the data supplied in the MIT/BIH Arrhythmia Database, we had to perform some data preprocessing. In achieving this task, the data associated with a recording is stored as a single vector for each individual heartbeat across all of the recorded readings. As part of each record, a label identifies the correct type of the heartbeat (e.g. normal, bradycardia or tachycardia) as shown on Table II based on the AAMI's recommendations. We extracted a total of eight from the dataset provided in MIT/BIH Arrhythmia's Database. Each heartbeat is stored as a vector containing the values of the corresponding eight features (or elements). A ninth element in the vector indicates the label or truth annotation.

A conventional method for classifying a heartbeat is based on considering an R-wave for the current ECG signal and a former R-wave (e.g. the earlier heartbeat's R-wave). In an effort to maximize the efficiency of this conventional method, we extend this method by considering the relationships of the heartbeats with earlier ones but also ones that occur before earlier heartbeats (i.e. a window that spans across two

TABLE II
Heartbeat Classification Based On The Number Of Beats Per Minute

Type	Identifier	Beats Per Minute (BPM)
Sinus Rhythm	1	sixty – one hundred
Bradycardia	2	< sixty
Tachycardia	3	> one hundred

heartbeats backward). Similarly, the relationship between the current beat and the next two beats (i.e. a window of two beats afterward) is taken into consideration. This enables the construction of the features including a wider combination of chronological RR-intervals.

The selection and extraction of the eight ECG features from the dataset provided by the MIT/BIH Arrhythmia Database include temporal (or non-stationary) and morphological properties. The temporal properties include: (a) the RR-interval between a heartbeat being examined (an existing heartbeat) and a preceding heartbeat (referred to as RR1), (b) between the preceding heartbeat and a former heartbeat (referred to as RR0), (c) between the existing heartbeat and the next heartbeat (referred to as RR2), and (d) between the next heartbeat and the subsequent heartbeat (referred to as RR3). An abnormality ratio is also generated based on the collected RR-intervals that include the (a) ratios of RR1 to RR0 and (b) RR3 to RR2. These ratios are used as indicators of heartbeat abnormalities particularly in identifying a normal heart activity (i.e. sinus rhythm), a slow heartbeat activity (i.e. bradycardia) or fast heartbeat activity (i.e. tachycardia).

The other group represents morphological properties represent. For example, an ECG signal is acquired as a point window consisting of two hundred and sixteen points in total. The label is used as an indication of the highest amplitude (or index) or the R peak in QRS complexes. A total of seventy six points preceding the R peak we considered as well as a total of one hundred and thirty nine points were used to form a template or window. All points in a given template vector are then normalized using min-max normalization presented in Equation 1. The result is a normalized vector ranging between zero and one.

$$v' = \left(\frac{v - min}{max - min}\right) \qquad (1)$$

where v' represents a normalized vector, v represents an un-normalized vector, min is the smallest value in v and max is the largest value in v.

Comparing a heartbeat that is being analyzed to preceding heartbeats provides important information such as pattern similarity. To this extent, the last two values in vector v represent the correlation of a heartbeat currently being analyzed with the two preceding ones, respectively. As part of preparing the data for further analysis, the entire heartbeats included in the dataset were obtained for further analysis while ignoring the first and last heartbeat accounting for setup and calibration that may take place when attaching or detaching equipment on the patient's body.

III. RESULTS & DISCUSSION

The dataset used in this study is extracted from thirty three patients from the MIT/BIH Arrhythmia Database. A patient's recording spanned across thirty minute duration. The three types of arrhythmias listed in Table II were considered including Sinus Rhythm, Bradycardia and Tachycardia. A total of 74182 total ECG readings were obtained from the dataset. Each reading or sample represents a single ECH heartbeat consisting of a template vector of nine features. A class label (or truth value) is represented by the last element in this vector. We obtained the truth value (or class label) from the annotations also supplied by the data source provider. This truth value has been identified by cardiologists and is referred to as the truth annotations. We use this label to train and evaluate our BPNN solution. To determine the effectiveness of using a BPNN in classifying ECG heartbeats into one of the three classification types, we use the Equation 5 for measuring classification rate (CR).

$$Classification\ Rate\ (CR) = 100 \times \left(\frac{Correctly\ Classified}{Total\ Heartbeats}\right) \qquad (5)$$

where CR represents the classification rate. We tested the BPNN which generated the results presented in Table III.

TABLE III
TESTING BPNN WITH DIFFERENT CONFIGURATIONS VARYING NUMBER OF HIDDEN NODES

Number of Nodes			CR
Input Layer	Hidden Layer	Output Layer	
9	1	1	80.565
9	2	1	83.335
9	3	1	81.781
9	4	1	83.849
9	5	1	80.645
9	6	1	82.568
9	7	1	86.319
9	8	1	83.847
9	9	1	91.019
9	10	1	86.487
9	11	1	85.295
9	12	1	91.347
9	13	1	97.745
9	**14**	**1**	**99.124**
9	15	1	96.359
9	18	1	93.891

Applying a wide variety of dissimilar network configuration, we determined that the BPNN configuration having fourteen hidden nodes is one that yields the highest classification rate (CR) with 99.124%. We applied the BPNN configuration having fourteen hidden nodes to train all of the samples in the dataset, or the 74182 heartbeats (or readings). We used a subset of the readings for testing. In particular, we used 53388 readings out of the 74182 (or 72%) for testing in which a classification rate of 98.70% was achieved. A pie chart showing the distribution of the correctly and incorrectly classified heartbeats based on the recommended BPNN configuration is shown in Fig. 1.

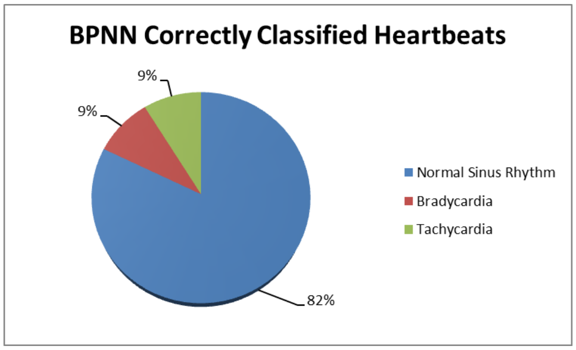


Fig. 1. Distribution of the Correctly Classified ECG Beats using BPNN

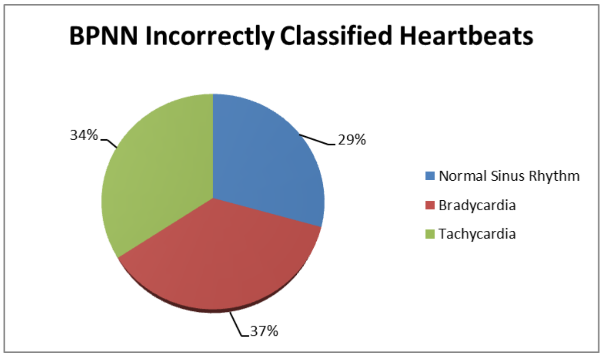


Fig. 2. Distribution of the Incorrectly Classified ECG Beats using BPNN

As presented in Fig.1, the total number of heartbeat readings that were correctly classified across all arrhythmia types is 52692 heartbeats of a total of 5388 heartbeats. This represents a correct classification rate of 98.70% across all types. Furthermore, it is noted that total number of correctly classified heartbeats for both tachycardia and bradycardia is equal percentage wise. A pie chart distribution chart showing the incorrectly classified heartbeats is shown in Fig. 2.

Fig. 2 shows, to a certain degree, a comparable distribution of the incorrectly classified heartbeats for both bradycardia and tachycardia with percentages being 37% and 34%, respectively. A slightly lower percentage is associated with the normal sinus rhythm. This slight variation between normal and abnormal heartbeats can be attributed to the complexity of determining normal heartbeats (i.e. normal sinus rhythm) and abnormal heartbeats (i.e. bradycardia and tachycardia).

It is important to note that the rate of occurrence in which the number of incorrect classification from normal as abnormal heartbeat is approximately twice that of abnormal as normal. For example, the number of incorrectly classified beats of normal sinus rhythm as Bradycardia is 207 while the opposite is 106 beats. The same observation applies to the comparison between tachycardia and normal sinus rhythm and is illustrated in the three-dimensional graph shown on Fig. 3.

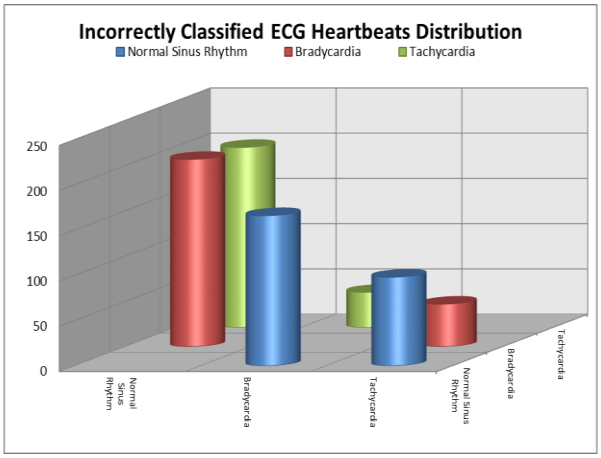


Fig. 3. Incorrectly Classified Distribution ECG Heartbeats

Converging of a BPNN is now always guaranteed in finding an error that is lower than the minimum during the training process. Hence, we used an acceptable error value of 1/2000. Although this may increase the time it takes for the BPNN to converge, we use this acceptable error value to achieve moderate classification rates. However, BPNN's ability to account for preceding delta of an earlier connection between the layers, the BPNN algorithm converges much quicker while taking into consideration the momentum compared to that without the momentum.

IV. CONCLUSION

In this paper, we presented a feedforward backpropagation neural network that can correctly classify abnormalities in heartbeats. In particular, this BPNN system is used for arrhythmias detection. We used MIT/BIH Arrhythmia Database as the data source. The properties obtained from this data source mainly focuses on the RR-intervals for various heartbeats. Results obtained from our testing demonstrate significant improvements in terms of the system's performance and correctly classifying ECG signals when considering heartbeats other than the current heartbeat (i.e. two previous heartbeats).

The average performance of arrhythmia classification is shown to be 98.70% and is comparable to other approaches that used the same data source. The high performance of the current system was tested against the set of features chosen and results show that the addition of features such as examining two beats ahead and prior to the current beat being examined improves the overall system performance

REFERENCES

[1] B. N. Hung, Y. S. Tsai, and T. H. Chu, "FFT algorithm for PVC detection using IBM PC," in Proc. IEEE Eng. Med. and Biol. Soc. 8th Annual International Conference, pp. 292-295, 1986.

[2] B. N. Hung, H. F. Cheng, and Y. S. Tsai, “An application of fast Walsh transform in ECG diagnosis,” in Proc. IEEE Eng. Medical and Biology Society 9th Annual. International Conference, pp. 497-498, 1987.

[3] J. Nadal and R. B. Panerai, “Classification of cardiac arrhythmias using principal component analysis of the ECG,” in Proceedings of IEEE Eng. Medical and Biology Society 13th Annual. International Conference, pp. 580-581, 1991.

[4] K. P. Lin and W. H. Chang, “QRS feature extract ion using linear prediction,” IEEE Trans. Biomedical. Engineering, vol. 36, no. 10, pp. 1050-1055, 1989.

[5] W. H. Chang, K. P. Lin, and S. Y. Tseng, “ECG analysis based on Hilbert transform descriptor,” in Proc. IEEE Eng. Med. and Biol. Soc. 10th Annual. International Conference, pp. 36-37, 1988.

[6] Yu, Liu, and Lee, “Hilbert transform in computer electrocardiographic diagnosis,” J. BMES R.O.C., vol. 5, no. 3, pp. 39-54, Sept. 1985.

[7] N. V. Thakor and Y. S. Zhu, “Applications of adaptive filtering to ECG analysis: Noise cancellation and arrhythmia detection,” IEEE Trans. Biomedica. Eng., vol. 38, no. 8, pp. 785-794, Aug. 1991.

[8] L. Sommo, P. 0. Borjesson, ‘M. E. Nygards, and 0. Pahlm. “A method for evaluation of QRS shape features using a mathematical model for the ECG,” IEEE Trans. Biomedical Engineering, vol. 28, pp. 713-717, 1981.

[9] D. A. Coast, R. M. Stem, G. G. Cano, and S. A. Briller, “An approach to cardiac arrhythmia analysis using hidden Markov models,” IEEE Trans. Biomed, Eng., vol. 37, pp. 826-836, Sep 1990.

[10] S. Osowsaki and T. H. Linh, “ECG beat recognition using fuzzy hybrid neural network,” IEEE Trans. Biomed. Engineering, vol. 48, pp. 1265-1271, Nov 2001.

[11] Y. H. Hu, S. Palreddy, and W. Tompkins, “A patient adaptable ECG beat classifier using a mixture of experts approach,” IEEE Trans. Biomed. Eng., vol. 44, pp. 891-900, Sept 1997.

[12] M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandt, and L. Sommo, “Clustering ECG complexes using Hermite functions and self-organizing maps,”, IEEE Trans. Biomed. Eng., 1998

[13] Y. Sun, “Arrhythmia recognition from electrocardiogram using non-linear analysis and unsupervised clustering techniques,” Ph.D. dissertation at Nanyang Technological University, 2001.

[14] K. Minami, Y. Ohkuma, H. Nakajima, and T. Toyoshima, “Arrhythmia diagnosis system which can distinguish atrial arrhythmias from ventricular rhythms,” in 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Amsterdam, pp. 1640-1641, 1996.

[15] K. Minami, H. Nakajima, and T. Toyoshima, “Rea l-time discrimination of ventricular tachyarrhythmia with Fourier-transform neural network,” IEEE Trans. Biomed. Eng., vol. 46, no. 2, Feb 1999.

[16] P. Chazal and R. B. Reilly, “A comparison of the use of different wavelet coefficients for the classification of the electrocardiogram,”. in 15th Conference on Pattern Recognition, vol. 2, pp. 255-258, 2000.

[17] K. P. Lin and W. H. Chang, “Classification of QRS pattern by an associative memory model,” in Proc. IEEE Eng. Med. and Biol. Soc. 430 IEEE 11th Annual International. Conference, pp. 2017-2018, 1989.

[18] J. Y. Cheung and S. S. Hull, Jr., “Detection of abnormal electrocardiograms using a neural network approach,” in Proc. IEEE Eng. Med. And Biol. Soc. 11th Annual International Conference, pp. 2015-2016, 1989.

[19] E. Pietka, “Neural nets for ECG classification,” Proc. IEEE Eng. Med and Biol. Soc., pp. 2021-2022, 1989.

[20] Massachusetts Institute of Technology (MIT) and Beth Israel Hospital (MIT/BIH) Arrhythmia Database, “http://ecg.mit.edu.”

[21] Q. Xue, Y. H. Hu, and W. J. Tompkins, “Neural-network-based adaptive matched filtering for QRS detection,” IEEE Trans. Biomed. Eng., vol. 39, no. 4, pp. 317-329, Apr. 1992.

[22] American Heart Association Scientific Databases, http://www.americanheart.org, Last Accessed Jan 28, 2018.

[23] Association for the Advancement of Medical Instrumentation (AAMI), http://www.aami.org, Last Accessed Jan 28, 2018.

[24] H Kang-Ping Lin and W. H. Chang, "A technique for automated arrhythmia detection of Holter ECG," Proceedings of 17th International Conference of the Engineering in Medicine and Biology Society, Montreal, Que., 1995, pp. 183-184 vol.1.

[25] P. Shimpi, S. Shah, M. Shroff and A. Godbole, "A machine learning approach for the classification of cardiac arrhythmia," 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 603-607, 2017.

[26] H. I. Bulbul, N. Usta and M. Yildiz, "Classification of ECG Arrhythmia with Machine Learning Techniques," 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, pp. 546-549, 2017.

[27] K. Suotsalo and S. Särkkä, "Detecting malignant ventricular arrhythmias in electrocardiograms by Gaussian process classification," 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP), Tokyo, pp. 1-5, 2017.

[28] V. Kalidas and L. S. Tamil, "Enhancing accuracy of arrhythmia classification by combining logical and machine learning techniques," Computing in Cardiology Conference (CinC), Nice, pp. 733-736, 2015.

[29] P. Cp, A. Suresh and G. Suresh, "Prediction of cardiac arrhythmia type using clustering and regression approach (P-CA-CRA)," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 51-54, 2017.

[30] S. Fallet, S. Yazdani and J. M. Vesin, "A multimodal approach to reduce false arrhythmia alarms in the intensive care unit," 2015 Computing in Cardiology Conference (CinC), Nice, pp. 277-280, 2015.

[31] T. W. Ow, W. Y. Chia, R. Bakhteri and Y. W. Hau, "SoC-based design of arrhythmia detector," 2014 2nd International Conference on Electronic Design (ICED), Penang, pp. 42-46, 2014.

[32] R. Ahmed and S. Arafat, "Cardiac arrhythmia classification using hierarchical classification model," 2014 6th International Conference on Computer Science and Information Technology, pp. 203-207, 2014.

[33] Y. Alwan, Z. Cvetković and M. J. Curtis, "High-dimensional discriminant analysis of human cardiac arrhythmias," 21st European Signal Processing Conference (EUSIPCO 2013), pp. 1-5, 2013.

[34] V. P. Nambiar, M. Khalil-Hani and M. N. Marsono, "Evolvable Block-based Neural Networks for real-time classification of heart arrhythmia From ECG signals," 2012 IEEE-EMBS Conference on Biomedical Engineering and Sciences, Langkawi, pp. 866-871, 2012.

[35] J. Igual and R. Llinares, "A new constrained Independent Component Analysis method to analyze atrial arrhythmias," 2011 7th International Symposium on Image and Signal Processing and Analysis (ISPA), Dubrovnik, pp. 552-557, 2011.

[36] S. M. Jadhav, S. L. Nalbalwar and A. A. Ghatol, "ECG arrhythmia classification using modular neural network model," 2010 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES), Kuala Lumpur, pp. 62-66, 2010.

[37] J. A. Nasiri, M. Sabzekar, H. S. Yazdi, M. Naghibzadeh and B. Naghibzadeh, "Intelligent Arrhythmia Detection Using Genetic Algorithm and Emphatic SVM (ESVM)," 2009 Third UKSim European Symposium on Computer Modeling and Simulation, Athens, pp. 112-117, 2009.

[38] S. R. Rogal Jr, A. B. Neto, M. Vinícius, M. Figueredo, E. C. Paraiso and C. A. A. Kaestner, "Automatic Detection of Arrhythmias Using Wavelets and Self-Organized Artificial Neural Networks," 2009 Ninth International Conference on Intelligent Systems Design and Applications, Pisa, pp. 648-653, 2009.

[39] A. Armato et al., "An FPGA Based Arrhythmia Recognition System for Wearable Applications," 2009 Ninth International Conference on Intelligent Systems Design and Applications, Pisa, pp. 660-664, 2009.

[40] A. Kuzmin, M. Mitrokhin, N. Mitrokhina, M. Rovnyagin and A. Alimuradov, "Intelligent data processing scheme for mobile heart monitoring system," 2017 XX IEEE International Conference on Soft Computing and Measurements (SCM), 2017, pp. 571-573.

[41] O. Bebek and M. C. Cavusoglu, "Model Based Control Algorithms for Robotic Assisted Beating Heart Surgery," 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, New York, NY, 2006, pp. 823-828.

[42] P. Couto, R. Ramalho and R. Rodrigues, "Suppression of False Arrhythmia Alarms Using ECG and Pulsatile Waveforms," 2015 Computing in Cardiology Conference (CinC), Nice, 2015, pp. 749-752.

.

RELATED CONFERENCES, CALL FOR PAPERS/PARTICIPANTS

TCII Sponsored Conferences

WI 2019
The 2019 IEEE/WIC/ACM International Conference on Web Intelligence
Thessaloniki, Greece
Oct. 20 - 23, 2019

Web has evolved as an omnipresent system which highly impacts science, education, industry, and everyday life. Web is now a vast data production and consumption platform at which threads of data evolve from multiple devices, by different human interactions, over worldwide locations under divergent distributed settings. Such a dynamic complex system demands adaptive intelligent solutions, which will advance know-ledge, human interactions and innovation. Web intelligence is now a cutting edge area which must address all open issues towards deepening the understanding of all Web's entities, phenomena, and developments.

The theme for the WI'19 is "Web Intelligence = AI in the Connected World". The 18th Web Intelligence conference (WI'19) aims to achieve a multi-disciplinary balance between research and technological disruptive advances in the fields of how intelligence is impacting the Web of people, The Web of Data, the Web of things, the Web of Trust, and The Web of health. WI'19 welcomes research, application as well as Industry/Demo track paper submissions in these core thematic pillars under a wide topics spectrum, which demand WI disruptive solutions for any of the next indicative sub-topics.
for the next generation of Web-empowered products, systems, services, and activities.

The Program Chairs of WI'19 are soliciting contributed technical papers for presentation at the Conference and publication in the Conference Proceedings by ACM.

ICDM 2019
IEEE International Conference on Data Mining
New York, USA
July 17-21, 2019
http://www.data-mining-forum.de/

The IEEE International Conference on Data Mining series (ICDM) has established itself as the world's premier research conference in data mining. It provides an international forum for presentation of original research results, as well as exchange and dissemination of innovative, practical development experiences. The conference covers all aspects of data mining, including algorithms, software and systems, and applications. ICDM draws researchers and application developers from a wide range of data mining related areas such as statistics, machine learning, pattern recognition, databases and data warehousing, data visualization, knowledge-based systems, and high performance computing. By promoting novel, high quality research findings, and innovative solutions to challenging data mining problems, the conference seeks to continuously advance the state-of-the-art in data mining. Besides the technical program, the conference features workshops, tutorials, panels.

ICHI 2019
The Seventh IEEE International Conference on Healthcare Informatics
Beijing China
June 10-13, 2019
http://www.ieee-ichi.org/

The Seventh IEEE International Conference on Healthcare Informatics (ICHI'19) will take place in Beijing from June 10th to June 13th, 2019 at Beijing International Convention Center.

The ICHI series is the premier community forum concerned with the application of computer science principles, information science principles, information technology, and communication technology to address problems in healthcare, public health, and everyday wellness. The conference highlights the most novel technical contributions in computing-oriented health informatics and the related social and ethical implications.

ICHI'19 serves as a venue for discussion of innovative technical and empirical contributions, highlighting end-to-end applications, systems, and technologies, even if available only in prototype form. ICHI'19 will feature keynotes, a multi-track technical program including papers, demonstrations, panels, workshop, tutorials, industrial tracks, and a doctoral consortium.

BigData 2019
The IEEE 2019 8th International Congress on Big Data (BigData Congress 2019)
Milan, Italy
July 8-13, 2019
http://conferences.computer.org/bigdatacongress/2019/

As cloud computing turning computing and software into commodity services, everything as a service in other words, it leads to not only a technology revolution but also a business revolution. Insights and impacts of various types of services (infrastructure as a service, platform as a service, software as a service, business process as a service) have to be re-examined.

2019 International Congress on Big Data (BigData Congress 2019) aims to provide an international forum that formally explores various business insights of all kinds of value-added "services." Big Data is a key enabler of exploring business insights and economics of services.

BigData Congress 2019's major topics include but not limited to: Big Data Architecture, Big Data Modeling, Big Data As A Service, Big Data for Vertical Industries (Government, Healthcare, etc.), Big Data Analytics, Big Data Toolkits, Big Data Open Platforms, Economic Analysis, Big Data for Enterprise Transformation, Big Data in Business Performance Management, Big Data for Business Model Innovations and Analytics, Big Data in Enterprise Management Models and Practices, Big Data in Government Management

Models and Practices, and Big Data in Smart Planet Solutions.

Related Conferences

AAMAS 2019
The 18th International Conference on Autonomous Agents and Multi-Agent Systems

Montreal
May 13-17, 2019
http://aamas2019.encs.concordia.ca/

AAMAS (International Conference on Autonomous Agents and Multiagent Systems) is the largest and most influential conference in the area of agents and multiagent systems. The aim of the conference is to bring together researchers and practitioners in all areas of agent technology and to provide a single, high-profile, internationally renowned forum for research in the theory and practice of autonomous agents and multiagent systems. AAMAS is the flagship conference of the non-profit International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

AAMAS'19, the 18th edition of the conference, will be held on May 13-17 in Montreal and is part of the Federated AI Meeting (FAIM), with the other conferences being IJCAI, ICML, ICCBR and SoCS. Please see the preliminary schedule for more information.

The AAMAS conference series was initiated in 2002 in Bologna, Italy as a joint event comprising the 6th International Conference on Autonomous Agents (AA), the 5th International Conference on Multiagent Systems (ICMAS), and the 9th International Workshop on Agent Theories, Architectures, and Languages (ATAL).

Subsequent AAMAS conferences have been held in Melbourne, Australia (July 2003), New York City, NY, USA (July 2004), Utrecht, The Netherlands (July 2005), Hakodate, Japan (May 2006), Honolulu, Hawaii, USA (May 2007), Estoril, Portugal (May 2008), Budapest, Hungary (May 2009), Toronto, Canada (May 2010), Taipei, Taiwan (May 2011), Valencia, Spain (June 2012), Minnesota, USA (May 2013), Paris, France (May 2014), Istanbul, Turkey (May 2015), Singapore (May 2016) , São Paulo (2017) and Stockholm, Sweden (2018) .

____

AAAI 2019
The 33rd AAAI Conference on Artificial Intelligence

Hilton Hawaiian Village, Honolulu, Hawaii, USA
January 27-February 1, 2019
https://aaai.org/Conferences/AAAI-19/

The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI'19) will be held January 27-February 1, 2019 at the Hilton Hawaiian Village, Honolulu, Hawaii, USA. The program chairs will be Pascal Van Hentenryck (Georgia Institute of Technology, USA) and Zhi-Hua Zhou (Nanjing University, China).

The purpose of the AAAI conference is to promote research in artificial intelligence (AI) and scientific exchange among AI researchers, practitioners, scientists, and engineers in affiliated disciplines. AAAI'19 will have a diverse technical track, student abstracts, poster sessions, invited speakers, tutorials, workshops, and exhibit and competition programs, all selected according to the highest reviewing standards. AAAI'19 welcomes submissions on mainstream AI topics as well as novel crosscutting work in related areas.

____

SDM 2019
The 2019 SIAM International Conference on Data Mining

Hyatt Regency Calgary | Calgary, Alberta, Canada
May 2 - 4, 2019
https://www.siam.org/Conferences/CM/Main/sdm19/

Data mining is the computational process for discovering valuable knowledge from data – the core of modern Data Science. It has enormous applications in numerous fields, including science, engineering, healthcare, business, and medicine. Typical datasets in these fields are large, complex, and often noisy. Extracting knowledge from these datasets requires the use of sophisticated, high-performance, and principled analysis techniques and algorithms. These techniques in turn require implementations on high performance computational infrastructure that are carefully tuned for performance. Powerful visualization technologies along with effective user interfaces are also essential to make data mining tools appealing to researchers, analysts, data scientists and application developers from different disciplines, as well as usable by stakeholders.

SDM has established itself as a leading conference in the field of data mining and provides a venue for researchers who are addressing these problems to present their work in a peer-reviewed forum. SDM emphasizes principled methods with solid mathematical foundation, is known for its high-quality and high-impact technical papers, and offers a strong workshop and tutorial program (which are included in the conference registration). The proceedings of the conference are published in archival form, and are also made available on the SIAM web site.

____

IJCAI 2019
The 28th International Joint Conference on Artificial Intelligence

Macao, China
August 10-16
http://www.ijcai19.org/

International Joint Conferences on Artificial Intelligence is a non-profit corporation founded in California, in 1969 for scientific and educational purposes, including dissemination of information on Artificial Intelligence at conferences in which cutting-edge scientific results are presented and through dissemination of materials presented at these meetings in form of Proceedings, books, video recordings, and other educational materials. IJCAI consists of two divisions: the Conference Division and the AI Journal Division. IJCAI conferences present premier international gatherings of AI researchers and practitioners and they were held biennially in odd-numbered years since 1969.

Starting with 2016, IJCAI conferences are held annually. IJCAI'19 will be held in Macao, P.R. China from August 10-16, 2019, IJCAI-PRICAI'20 in Nagoya, Japan, IJCAI'21 in Montreal, Canada and IJCAI-ECAI'22 in Bologna, Italy.

Special Issue on IEEE Intelligent Informatics Bulletin

Data-driven Intelligent Healthcare and Medicine

Abstract

The prospect of Personalized Precision Medicine (PPM), driven by Data Science (machine learning algorithms) and Artificial Intelligence (AI) applications, in tailoring the diagnostics used in individualized treatments to improve patient health and outcomes is very promising. Digital Health, as part of AI and PPM combination, will revolutionize healthcare delivery, optimize personalized and precision medicine, and offer new tools for drug and diagnostic development.

AI solutions can be implemented by using knowledge represented as patterns and rules extracted from data processed via machine learning. PPM approach for disease treatment and prevention include individual variability in genes, environment and lifestyle for each person; linking large scale of genomics, other omics, biomedical imaging with a large scale of electronic patient health care records and e-record creating a big data set. Digital Health applications as biosensors, mobile devices and wearables, mobile health platforms and digital biomarkers are quickly expanding into all areas of patient monitoring and disease management, point-of-care diagnostics, and digital end points in clinical trials. Also, this integrative approach empowers the patients (smart patients) and convert the big amount of raw data into Smart Data by combining volume, velocity, variety, and veracity.

The proposed Special Issue is an effort and a step taken forward to explore PPM, driven by data science and AI applications to improve patient health and outcomes including digital health. The discussions will encompass the theoretical basis and related tools to formally represent, measure, model, and mine meaningful patterns from large-scale medicine datasets related to AI, WWW, and Computational Social Science. Specifically, the objectives of the proposed Special Issue are:

- To discuss on challenging issues in PPM, driven by Data Science and AI applications including Digital Health seeking the related breakthrough for new revolutions in related fields;
- To explore an innovative route to smartly merge technologies in related fields and make online smart data more productive and intelligent to the institutes related with health;
- To provide a forum for researchers to discuss their recent work on the topics mentioned above and facilitate research collaboration and potential research projects and development directions in technologies, methodology, and applications from different countries and regions and for all the Intelligent Informatics Community

Proposed Timeline

Manuscript submission deadline:	28 February 2019
First round notification with reviewer comments:	31 March 2019
Second round submission:	28 April 2019
Final acceptance notification:	19 May 2019

Guest Editors

- Dr. María Flavia Guiñazu. Chatham House UK Web Intelligence Centre, University of Chile, Chile. Email: flavia.guinazu@wic.uchile.cl
- Dr. Xiaohui Tao. University of Southern Queensland (USQ), Australia. Email: xiaohui.tao@usq.edu.au
- Dr. Juan D. Velásquez Web Intelligence Centre, University of Chile, Chile. Email: jvelasqu@dii.uchile.cl

IEEE Computer Society
1730 Massachusetts Ave, NW
Washington, D.C. 20036-1903