INTERNATIONAL JOURNAL
of
COMPUTERS COMMUNICATIONS & CONTROL

ISSN 1841-9836, e-ISSN 1841-9844

A Bimonthly Journal
With Emphasis on the Integration of Three Technologies

Year: 2019 Volume: 14 Issue: 4 Month: August

This journal is a member of, and subscribes to the principles of, the Committee on Publication Ethics (COPE).

C O P E
Member since 2012
JM08090

http://univagora.ro/jour/index.php/ijccc/

CCC Publications

BRIEF DESCRIPTION OF JOURNAL

Publication Name: International Journal of Computers Communications & Control.
Acronym: IJCCC; **Starting year of IJCCC:** 2006.
ISO: Int. J. Comput. Commun. Control; **JCR Abbrev:** INT J COMPUT COMMUN.
International Standard Serial Number: ISSN 1841-9836, e-ISSN 1841-9844.
Publisher: CCC Publications - Agora University of Oradea.
Publication frequency: Bimonthly: Issue 1 (February); Issue 2 (April); Issue 3 (June); Issue 4 (August); Issue 5 (October); Issue 6 (December).
Founders of IJCCC: Ioan DZITAC, Florin Gheorghe FILIP and Misu-Jan MANOLESCU.

Indexing/Coverage:

- Since 2006, Vol. 1 (S), IJCCC is covered by Clarivate Analytics and is indexed in ISI Web of Science/Knowledge: Science Citation Index Expanded.

  2019 Journal Citation Reports® Science Edition (Clarivate Analytics, 2018):
  Subject Category: (1) Automation & Control Systems: Q4(2009, 2011, 2012, 2013, 2014, 2015), **Q3(2010, 2016, 2017, 2018)**; (2) Computer Science, Information Systems: Q4(2009, 2010, 2011, 2012, 2015), **Q3(2013, 2014, 2016, 2017, 2018)**.

  Impact Factor/3 years in JCR: 0.373(2009), 0.650 (2010), 0.438(2011); 0.441(2012), 0.694(2013), 0.746(2014), 0.627(2015), 1.374(2016), 1.29 (2017), **1.585 (2018)**.
  Impact Factor/5 years in JCR: 0.436(2012), 0.622(2013), 0.739(2014), 0.635(2015), 1.193(2016), 1.179(2017), **1.361(2018)**.

- Since 2008 IJCCC is indexed by Scopus: **CiteScore2018 = 1.56**.
  Subject Category:
  (1) Computational Theory and Mathematics: Q4(2009, 2010, 2012, 2015), **Q3(2011, 2013, 2014, 2016, 2017, 2018)**;
  (2) Computer Networks and Communications: Q4(2009), Q3(2010, 2012, 2013, 2015), **Q2(2011, 2014, 2016, 2017, 2018)**;
  (3) Computer Science Applications: Q4(2009), **Q3(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018)**.

  SJR: 0.178(2009), 0.339(2010), 0.369(2011), 0.292(2012), 0.378(2013), 0.420(2014), 0.263(2015), 0.319(2016), 0.326 (2017), **0.37 (2018)**.

- Since 2007, 2(1), IJCCC is indexed in EBSCO.

Focus & Scope: International Journal of Computers Communications & Control is directed to the international communities of scientific researchers in computers, communications and control, from the universities, research units and industry. To differentiate from other similar journals, the editorial policy of IJCCC encourages the submission of original scientific papers that focus on the integration of the 3 "C" (Computing, Communications, Control).

In particular, the following topics are expected to be addressed by authors:

(1) Integrated solutions in computer-based control and communications;

(2) Computational intelligence methods & Soft computing (with particular emphasis on fuzzy logic-based methods, computing with words, ANN, evolutionary computing, collective/swarm intelligence, membrane computing, quantum computing);

(3) Advanced decision support systems (with particular emphasis on the usage of combined solvers and/or web technologies).

EDITORIAL STAFF OF IJCCC

EDITORIAL ADDRESS:
Agora University, Cercetare Dezvoltare Agora, Tineretului 8, 410526 Oradea, Bihor, Romania,
Tel./ Fax: +40 359101032, E-mail: ijccc@univagora.ro, rd.agora@univagora.ro
URL: http://univagora.ro/jour/index.php/ijccc/

EDITORIAL BOARD OF IJCCC (MEMBERS:

Contents

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 459-474, August 2019.

ESBL: Design and Implement A Cloud Integrated Framework for IoT Load Balancing

G. Balakrishna, M.N. Rao

Gubba Balakrishna
Koneru Lakshmaiah EducationFoundation
Department of Computer Science and Engineering
me2balu@gmail.com

Nageswara Rao Moparthi
Koneru Lakshmaiah Education Foundation
Department of Computer Science and Engineering
rao1974@gmail.com

Abstract: The continuous growth in wireless communication, the demand for sophisticated, simple and low-cost solutions are also increasing. The demand motivated the researchers to indulge into inventing suitable network solutions ranging from wireless sensor networks to wireless ad-hoc networks to Internet of Things (IoT). With the inventions coming from the researchers, the demand for further improvements into the existing researchers have also growth upbound. Initially the network protocols were the demand for research and further improvements. Nevertheless, the IoT devices are started getting used in various fields and started gathering a huge volume of data using complex application. This invites the demands for research on load balancing for IoT networks. Several research attempts were made to overcome the communication overheads caused by the heavy loads on the IoT networks. Theses research attempts proposed to manage the loads in the network by equally distributing the loads among the IoT nodes. Nonetheless, in the due course of time, the practitioners have decided to move the data collected by the IoT nodes and the applications processing those data in to the cloud. Hence, the challenge is to build an algorithm for cloud-based load balancer matching with the demands from the IoT network protocols. Hence, this work proposes a novel algorithm for managing the loads on cloud integrated IoT network frameworks. The proposed algorithm utilizes the analytics of loads on cloud computing environments driven by the physical host machines and the virtual environments. The major challenge addressed by this work is to design a load balancer considering the low availability of the energy and computational capabilities of IoT nodes but with the objective to improve the response time of the IoT network. The proposed algorithm for load balancer is designed considering the low effort integrations with existing IoT framework for making the wireless communication world a better place.

Keywords: IoT, cloud integrated load balancer, inter quartile correlation, static threshold utilization.

1 Introduction

The most recent ground-breaking innovation in the field of networking is Internet of Things (IoT). The IoT networks have gained a huge popularity for making the communication protocols working with three simple components as gather the data from the open operational environment using sensors, the independent machine to machine communication and the cloud computing environment for the applications and the data collected by the sensor agents. An important aspect of IoT is their protection against cyber attacks [1]. The increasing popularity of smart conceptual factors of modern appliances and life style made IoT an integral part of the modern wireless communications. IoT frameworks are frequently considered as a layered secluded engineering of a computerized innovation. The gadget layer alludes to the physical parts: CPS, sensors or machines. The system layer comprises of physical system transports, distributed computing and correspondence conventions that total and transport the information to the administration layer [3], which comprises of uses that control and consolidate information into data that can be shown on the driver dashboard. The best most stratum of the stack is the substance layer or the UI.

The historical backdrop of the IoT starts with the development of the programmable rationale controller by Dick Morley in 1968, which was utilized by GM in their programmed transmission producing division. These PLCs took into consideration fine control of individual components in the assembling chain. In 1975, Honeywell and Yokogawa presented the world's first DCSs, the TDC 2000 and the CENTUM, independently. These DCSs were the following stage in permitting adaptable process control all through a plant, with the additional advantage of reinforcement redundancies by conveying control over the whole framework, disposing of a solitary purpose of disappointment in a focal control room.

Nonetheless, with the increasing demand for further improvements on IoT load balancing for cloud integrated applications and frameworks this work proposes a novel load balancing algorithm.

The rest of the work is furnished such as in the section 2, the routing demands for any IoT network is analysed, in section 3, the benefits of cloud computing service models are analysed, in section 4, the outcomes from the parallel researches are analysed, in the section 5, the novel proposed load balancing algorithm is presented, in the section 6, the integration possibilities of the proposed algorithm into any existing IoT framework is discussed, in the section 7, the results obtained from the proposed framework is discussed and analysed, in the section 8, the comparative analysis with the standard protocols are carried out to establish the significance of improvements and this work presents the final conclusion of the research in the section 9.

2 IoT routing strategies demands

In this section of the work, the fundamental demands for routing strategies on IoT is elaborated and discussed to realise the break points for performance improvements using load balancing.

Some expansive scale remote information securing, and activation related applications utilize low-fuelled installed gadgets. These applications incorporate accuracy horticulture, building the board/mechanical robotization, vehicular specially appointed systems, and urban systems/vitality and water lattices to fabricate more intelligent urban communities. In these remote sensors organizes, the installed gadgets work under serious vitality limitations, which results in calculation, stockpiling, and radio-transmission related imperatives [Fig. 1]. They likewise impart over a lossy channel.

Traffic examples and information stream inside LLN are profoundly directional. The examples can be characterized as multipoint-to-point traffic, point-to-multipoint traffic, or point-to-point traffic. In MP2P traffic, for instance, detected data from various detecting hubs is steered to an Internet application by means of LBR. P2MP traffic is seen when an inquiry asks for is produced using the Internet and steered by means of the LBRs and LLN switches to various field hubs. P2P traffic happens when control data should be sent to actuator or ready data is gotten from a sensor.

The RFC reports depict the key usefulness and steering prerequisites for urban LLNs:

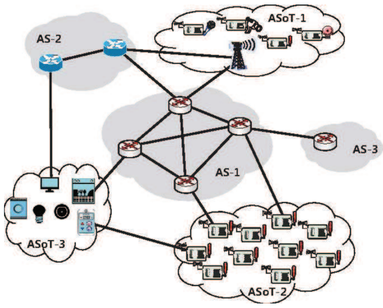


Figure 1: IoT routing building blocks

2.1 Arrangement of nodes

In a run of the mill urban system organization, hundreds or thousands of nodes with pre-customized functionalities are taken off. The steering convention should represent these variables and bolster self-association and self-design at the most reduced conceivable vitality cost.

2.2 Affiliation and disassociation of nodes

After the instatement stage, nodes may join or leave the system at self-assertive occasions. The steering convention likewise ought to most likely handle circumstances where a breaking down node may influence or endanger the general directing effectiveness.

2.3 Ordinary estimation detailing

The information directing calculation and choice may rely upon the detected information, the recurrence of revealing, the measure of vitality staying in the nodes, the reviving example of the vitality searched nodes, or different variables.

2.4 Scalability

The directing convention must almost certainly bolster a field organization of a couple of hundred to a huge number of sensor nodes without decaying chosen execution parameters beneath configurable edges.

Hence, it is natural to understand that the scalability is one of the key factors for low power consuming devices, such as IoT devices. Thus, the scalability of the IoT network can be improved using cloud-based load balancing strategies.

Thus, in the next section of the work, the cloud computing service types with related advantages are discussed.

3 Cloud computing services

This work proposes a novel framework to balance the IoT network data processing loads using the cloud computing services. Hence, it is a prime importance to realize the cloud computing service types and identify the benefits, which can be utilized for balancing IoT network loads.

3.1 Using cloud software capabilities

The ability gave to the data consumers is to utilize applications running on a cloud foundation. The applications are available from different gadgets through either a thin interface, for

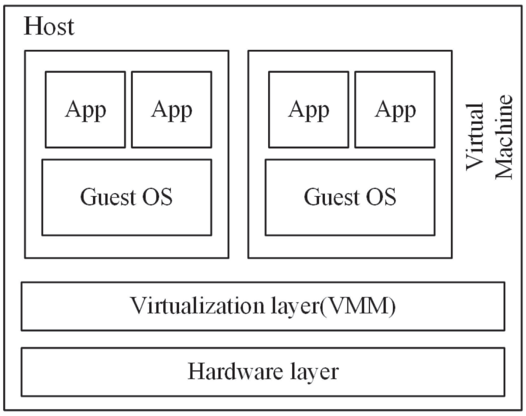


Figure 2: Cloud provisioning capabilities using VM

example, an internet browser, or a programable interface. The consumer does not oversee or control the hidden cloud framework including system, servers, working frameworks, stockpiling, or even individual application capacities, with the conceivable exemption of constrained client explicit application design settings.

3.2 Using cloud operational capabilities

The ability gave to the application consumers to send application onto the cloud framework, data centre owners made or gained applications made utilizing programming dialects, libraries, administrations, and apparatuses bolstered by the supplier. The application owner does not oversee or control the basic cloud framework including system, servers, working frameworks, or capacity, however has power over the conveyed applications and potentially design settings for the application-facilitating condition.

3.3 Using cloud provisioning capabilities

The capacity gave to the application owners to arrangement handling, stockpiling, systems, and other key registering assets where the application consumers can send and run self-assertive programming, which can incorporate working frameworks and applications. The application owner does not oversee or control the basic cloud foundation but rather has command over working frameworks, stockpiling, and conveyed applications; and potentially constrained control of select systems administration parts

Henceforward, it is natural to realize that the provisioning capabilities of cloud computing can be utilized to handle the loads on applications running and generating data from any IoT network.

The provisioning of resources on cloud can be achieved using virtual machines [Figure 2]. Virtual machines are operating frameworks or application situations that is introduced on programming, which emulates devoted equipment. The end client has indistinguishable experience on a virtual machine from they would have on committed equipment. Specific programming, called a hypervisor, copies the PC customer or server's CPU, memory, hard circle, arrange, and other equipment assets totally, empowering virtual machines to share the assets. The hypervisor can imitate numerous virtual equipment stages that are disengaged from one another, enabling virtual machines to run Linux and Windows Server operating frameworks on the equivalent fundamental physical host. Virtualization limits cost by lessening the requirement for physical equipment frameworks.

4 Outcomes from the parallel researches

The origination of the modern research on IoT started with the newer directions explored by the S. Oteafy et al. in [15]. The work [15] defines the possibilities of utilizing the existing frameworks and strategies from wireless sensor networks. Though, this work formulated the foundation of the research, the deficiencies identified in this work is the designed networks are highly generalized and cannot be extended to the modern day demands from IoT networks. Soon the demand for specified routing strategies are identified and Q. Le et al. in [9] have proposed the enhanced routing strategies for WSN matching with the demands of IoT networks. The shortcoming from this proposed algorithm was the missing component of balancing load. This shortcoming was restructured by the C. Petrioli et al. in [17].

With the initiations by C. Petrioli et al. [17], the newer challenges are identified, and multiple parallel research outcomes are also demonstrated. The work by J. Guo et al. [6] for managing the large-scale network and the work by H. Y. Kim et al. [8] for increasing the life time of the network have demonstrated significant improvements over the existing situations.

Yet another challenge of the present IoT networks is the IoT nodes are deployed for various purposes and must collected information from various sources, in various formats and in various time durations. Hence, managing the load for higher to lower data ranges is obviously a challenge and must be addressed. The work of S. M. A. Oteafy et al. [14] demonstrates the possibilities for self-adaptive or self-adjusting algorithm for balancing loads. The IoT frameworks are based on the standard network protocols and the standard network stack, thus for balancing loads, the capabilities can be utilized. This ideology was proposed by Di Marco et al. in [4] for managing the loads using the network stack functionalities.

During the routing of the data, load balancing being the prime concern, the energy efficiency is also an important factor. With the improvements of the routing algorithms, it is been observed that, the higher complexities of the routing algorithms are pooling a lot of energy for execution. The energy consumptions must be reduced in order to increase the life span of the IoT network.S. A. Alvi et al. in [2] proposes a guideline for framing the routing algorithms following the green computing policies.

Hence, with different recommendations from various researchers on various aspects of IoT frameworks, there was a significant demand to summarize concrete guidelines for IoT design, deployment and managements. The recommendations were well summarized by M. Wu et al. in their work [19].

In spite of number of several prominent outcomes from the research community on IoT, it was a great difficulty to influence the practitioners to use the IoT frameworks as the initial implementations are costly for changing into a newer dimension. After the work of D. Evans et [5] explaining the benefits of using IoT networks for various demands, a high amount of adaptations was observed. Within few months, the demand for IoT implementation was increased to a greater extend. Naturally, the research community was interested to find the details of implementation as to what extend the innovations were helping the practitioners. The survey on the findings was reported timely by L. Mainetti et al. in [12].

The report by L. Mainetti et al. [12] have uncovered a newer research problem for the research community. The IoT network frameworks were designrf to sustain for shorter period and cannot be maintained for a longer duration. Thus, the practitioners have raised a serious concern about the stability of the networks as further improvements or implementation can increase the cost to a greater extend. The solution to this problem was proposed by S. A. Karthikey et al. [7] for making the IoT networks fault tolerant and cost effective.

Further, the resource management schemes are also to be made economical. Many of the parallel research attempts have demonstrated the best use of economic planner to drive the resource management as demonstrated by T. Menouer et al. [13]

Building a resource clusters on cloud for IoT based networks also proven to be significant for load optimization as suggested by R. Peinl et al. in [16]. Nevertheless, these strategies are primarily applicable for private cloud setups, hence these strategies are criticised by a larger research community. The moderated strategy parallel to this is proposed by C. C. Tarek Menouer et al. [18] to overcome the challenges of isolated cloud. Further the work of C.C. Tarek Menouer et al. [18] is enhanced by Keqin Li et al. [10] by using the Multiple Heterogeneous Servers specialized for edge computing.

The working models of these proposed strategies are numerous as one of the most popular deployment is described by Yue Xu et al. in [20] for traffic management and prediction based on the proposed framework by Longjiang Li et al. in [11].

Nonetheless, the demands and purposes for the IoT networks have came a long way from the initial research outcomes. The present IoT networks are utilized for higher and higher capacitive situations. The data collected by the network nodes have also grown in volume and the applications handling the data must also be scaled. Hence, the practitioners have started utilizing the benefits from cloud computing. The research outcomes from the cloud computing have proposed a good number of solutions to balance the load for cloud specific applications handling the data from the cloud services. Nevertheless, the algorithms are not designed and not suitable for handling applications dealing with IoT data.

Henceforth, this work proposes a novel framework for balancing loads for cloud integrated IoT framework. The novel algorithm is elaborated in the next section of this work.

5 IoT cloud integrated load balancing algorithm

The proposed framework deploys an algorithm for migrating the highly loaded virtual machine to a lesser loaded destination physical host. The proposed algorithm utilizes 4 major properties of virtual machines handling IoT node data as CPU utilization, Power consumption or the energy, memory utilization and finally the SLA violation due to the overload.

Cloud stack adjusting is the way toward disseminating remaining tasks at hand and registering assets in a cloud processing condition. Load adjusting enables ventures to oversee application or remaining task at hand requests by dispensing assets among various PCs, systems or servers. Cloud stack adjusting includes facilitating the circulation of outstanding burden traffic and requests that live over the Internet. The algorithm is furnished here.

From the understanding of the parallel research outcomes, it is natural to realise that most of the load balancing algorithms have failed in justifying the IoT network load balancing demands as all the parameters involved in analysing the load of any virtual machine were given equal weightage. In the contradiction, this work proposes a novel weightage metric to justify the IoT load balancing demands as studied in the previous sections of this work.

Considering the limited power and processing capabilities of any IoT node in the network, this algorithm improves the energy efficiency and the processing capabilities to a greater extend. Also, the loading balancing process time complexity is reduced significantly.

The working capabilities of the proposed ESBL algorithm with existing IoT framework is elaborated in the next section of this work.

6 Proposed framework for cloud integrated IOT load balancing

In this section of work, the proposed framework to balance load for cloud integrated IoT network is elaborated. The proposed ESBL algorithm can get embedded in the IoT function

Algorithm 1 Energy sensitive for balancing load algorithm (ESBL)

```
Initialize the IoT network routing table
Initialize all nodes
Accumulate the data from the IoT nodes into Cloud
Instantiate the Virtual machines
Analyse the present load as VM[i]
For each virtual machine
a. Calculate CPU utilization as CPUV[i]
b. Calculate power consumption as PowerV[i]
c. Calculate Memory utilization as MemV[i]
d. Calculate SLA violation as SlaV[i]
For each VM[i]
a. calculate the LoadV[i] = (CpuV[i] * 0.4) + (PowerV[i] * 0.1) +(MemV[i] * 0.3) + (SlaV[i] * 0.2)
For LoadV
a. IF $LoadV[i] > LoadV[j]$
b. $High_V M \leftarrow LoadV[i]$
c. $Low_V M \leftarrow LoadV[j]$
$SourceVM \leftarrow High_V M$
$DestinationVM \leftarrow Low_V M$
$CalculateVM_S hutDown$
$CalculateEnergyV \leftarrow PowerV[DestinationVM]$
$MigrateVM[DestinationVM]$
```

process. The proposed modified framework is elaborated in Figure 3.

In the initial phase of the network operations, the network components must be started and generate the beacon signal as part of the initial bootstrap process. The IoT involves extending Internet connectivity beyond standard devices, such as desktops, laptops, smart phones and tablets, to any range of traditionally dumb or non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the Internet, and they can be remotely monitored and controlled. All the devices associated with the network must be up and running in this initial phase of the framework and the framework is also responsible for assuring connectivity between the components.

In the next phase of the framework, the network routing table must be updated to ensure the appropriate data migration to the cloud. A routing table uses a similar thought that one does when utilizing a guide in bundle conveyance. At whatever point a hub needs to send information to another hub on a system, it should initially realize where to send it. If the hub can't straightforwardly associate with the goal hub, it needs to send it through different hubs along an appropriate course to the goal hub. Most hubs don't Endeavor to make sense of which course may work; rather, a hub will send an IP bundle to an entryway in the LAN, which at that point chooses how to course the "bundle" of information to the right goal. Every door should monitor which approach to convey different bundles of information, and for this, it utilizes a routing table. A routing table is a database which monitors ways, like a guide, and uses these to figure out which approach to forward traffic. Entry ways can likewise share the substance of their routing table with different hubs asking for that data.

Furthermore, the network controller using this framework must migrate all the data collected by the network to be migrated to the cloud hosts. The applications running on the cloud hosts shall use these data for further purposes.

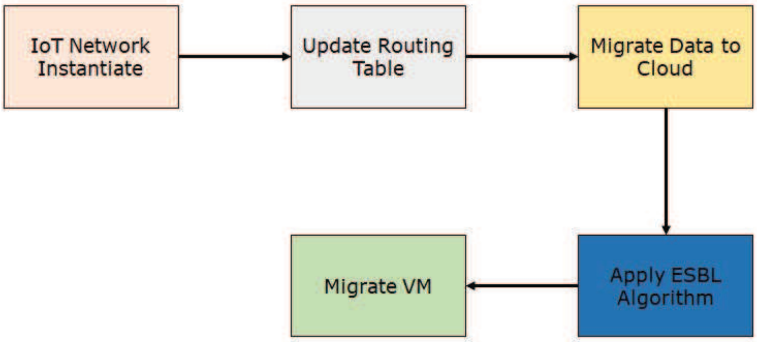


Figure 3: Proposed cloud integrated IoT application load balancing process flow

Table 1: Initial experimental setup

Algorithm Name	No. of hosts	Number of VMs	Total simulation time(sec)
IQR MMT	50	100	86400
IQR MU	50	100	86400
LR MMT	50	100	86400
LR MU	50	100	86400
MAD MMT	50	100	86400
MAD MU	50	100	86400
THR MMT	50	100	86400
THR MU	50	100	86400
ESLB	50	100	86400

Once the applications are up and running, the ESBL algorithm, will identify the heavily loaded source VM host instance and less loaded destination physical host. Finally, after the identification of source and destination nodes, the migration happens.

The framework continues to collect data from the IoT network and performs the load balancing strategies as and when imbalanced nodes are identified.

7 Results and discussions

The results obtained from the proposed algorithm and the cloud integrated framework are highly satisfactory. The obtained results are furnished in this section of the work and discussed further.

The results obtained from the proposed framework is analysed in few different parts as load analysis of the IoT network before and after the migration, Initial virtual machine setups, Energy consumption, Number of node shutdowns and finally Execution time of the total load balancing time. During the analysis of the results, the results obtained from the proposed algorithm is also compared with other parallel research algorithms for cloud-based load balancing.

7.1 Initial virtual machine setups

Firstly, the initial virtual machine setups are discussed in this section. All the algorithms, proposed and comparable, are tested under a similar situation. The initial setup is discussed in Table 1.

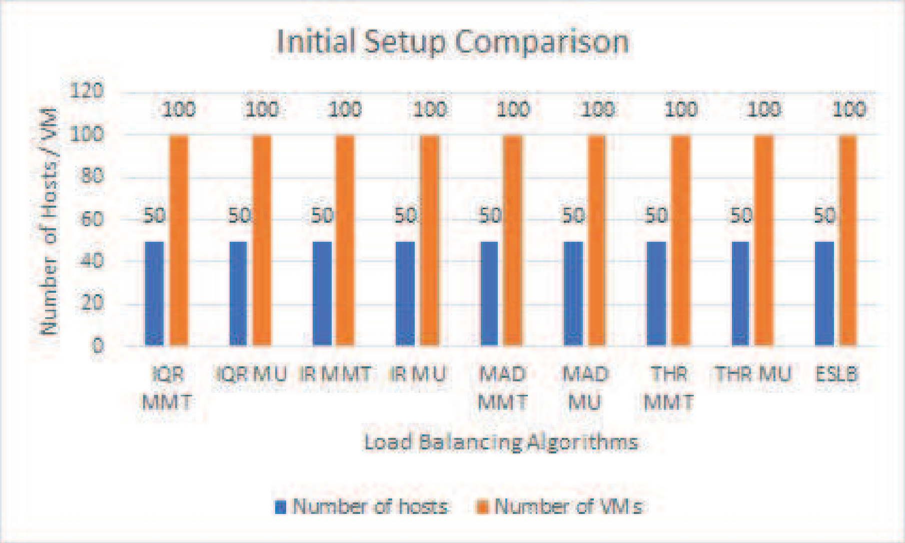


Figure 4: Initial setup comparisons

Table 2: Energy consumption analysis

Algorithm Name	Energy Consumption (KWh
IQR MMT	47.85
IQR MU	49.32
LR MMT	35.37
LR MU	35.38
MAD MMT	45.61
MAD MU	47.36
THR MMT	41.81
THR MU	44.08
ESLB	36.81

The detailed of the comparative algorithms are discussed in the further section of this work. The results are also visualized graphically in Figure 4.

7.2 Energy consumption

Secondly, the energy consumption by the virtual machines depend on the energy consumption for running the applications on the virtual machines and the load balancing. The calculation of the energy or power consumption is calculated for the complete virtual structure of the network, where all the virtual machines are included. This strategy helps in realizing the effective utilization of the power.

The load balancing algorithms are intended to reduce the load on one single physical instance or a specific virtual machine to reduce the power consumption. The power consumption of the proposed algorithm is analysed and compared with the other parallel standard load balancing algorithms [Table 2]. The results ate visualized graphically in Figure 5.

It is natural to understand that the overall energy consumption of the virtual network for IoT application and data management is significantly less compared to the other standard applications.

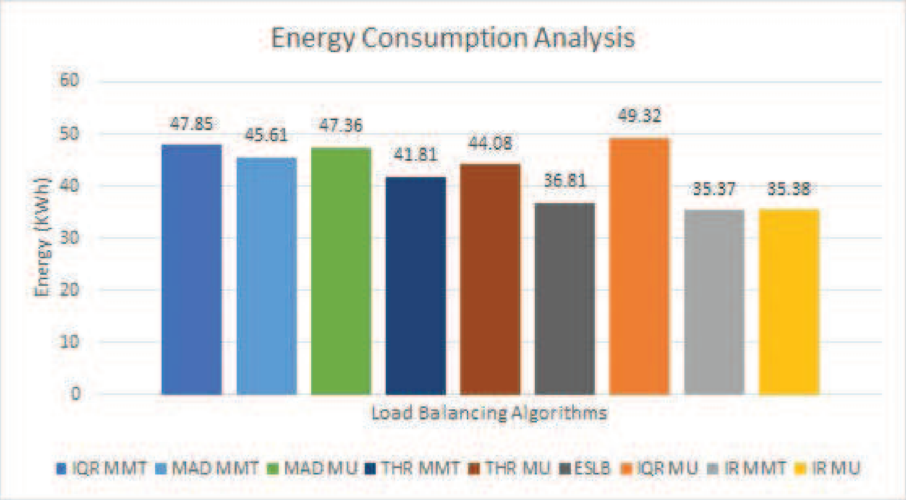


Figure 5: Energy consumption analysis

Table 3: Number of node (physical hosts) shutdowns

Algorithm Name	Number of node shutdowns
IQR MMT	1549
IQR MU	1622
LR MMT	806
LR MU	816
MAD MMT	1528
MAD MU	1632
THR MMT	1424
THR MU	1578
ESLB	795

7.3 Number of physical hosts shutdowns

Thirdly, the number of physical hosts during the idle phase tend to shutdown if for a longer duration no load is been assigned. Hence, the poorest load balancing algorithm will have maximum number of host shutdowns. The Physical hosts shutdown analysis is furnished in Table 3.

The results are been analysed graphically in Figure 6.

It is natural to realize that, the proposed algorithm is demonstrating much lesser hosts shutdowns, which clearly signifies that the utilization of the available resources is significantly high compared with the other algorithms.

7.4 Load balancing execution time

Fourthly, the load balancing algorithm time complexity is analysed. The code responsible for balancing the loads are also deployed on the same physical host instances and buy out the time required to run, or time required for application to respond. The time complexity is analysed in Table 4.

The results are been analyzed graphically in Figure 7.

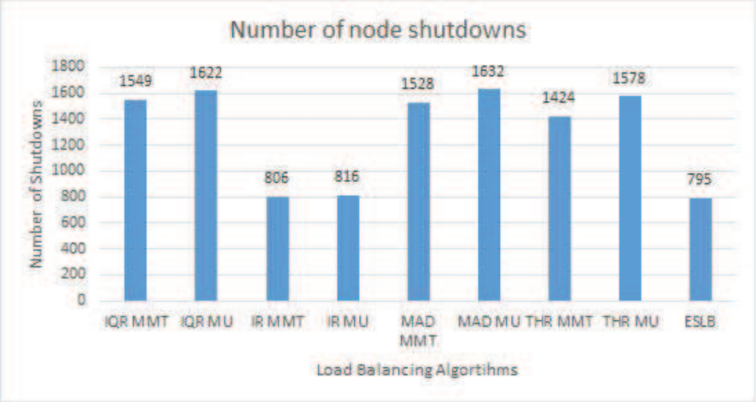


Figure 6: Host N node shutdown analysis

Table 4: Load balancing execution time

Algorithm Name	Execution time (Sec)
IQR MMT	0.002510
IQR MU	0.003100
LR MMT	0.001680
LR MU	0.001530
MAD MMT	0.003310
MAD MU	0.014710
THR MMT	0.001330
THR MU	0.001600
ESLB	0.001560

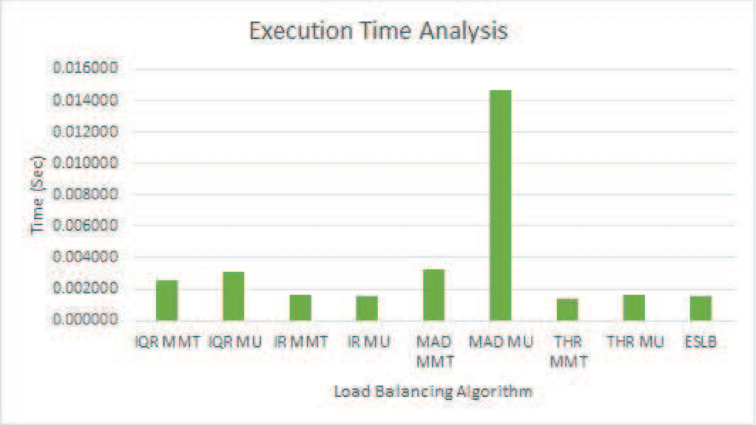


Figure 7: Execution time analysis

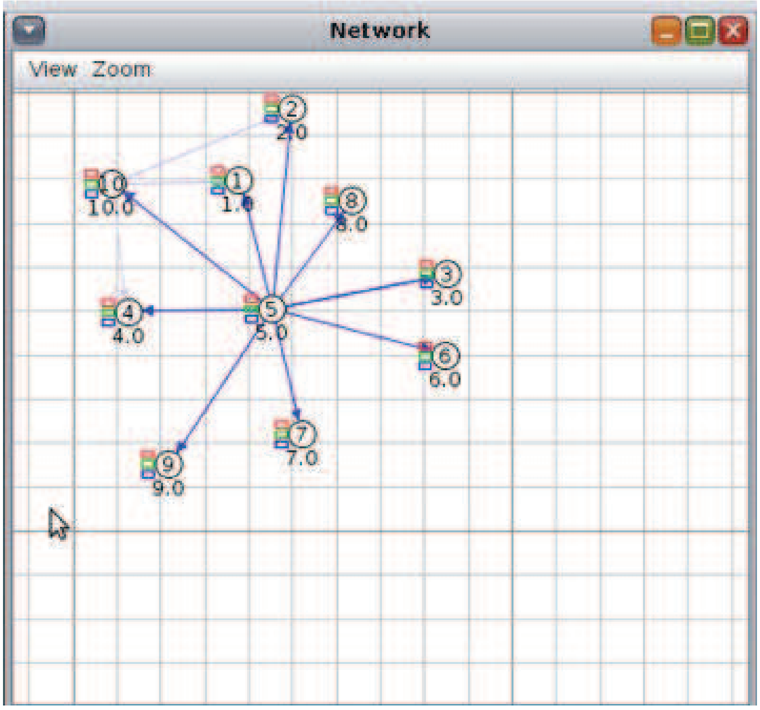


Figure 8: Initial insanitation of the network

7.5 IoT network performance

Finally, the performance of any IoT network can be measured by analysing the responsiveness of the network. During a network performance improvement process, if the response time or the number of response requests in the same time increases, then it is natural to accept that the performance of the network has improved.

The network is observed during the routing phases and some of the scenarios are furnished in Figures 8 – 10.

The performance comparison of the configured IoT network is analysed in Table 5.

Table 5: Load balancing effects on IoT network

Source Node	Number of Connect Requests	Average Response with ESBL	Average Response without ESBL
1	491	2.063	1.941
2	390	2.086	1.921
3	297	2.092	1.916
4	398	2.187	1.843
5	971	2.093	1.915
6	393	2.015	1.985
7	291	2.109	1.902
8	491	2.098	1.911
9	285	2.192	1.839
10	395	2.147	1.872

The improvement is visualized graphically in Figure 11.

It is natural to observe that the response time have significantly improved without compro-

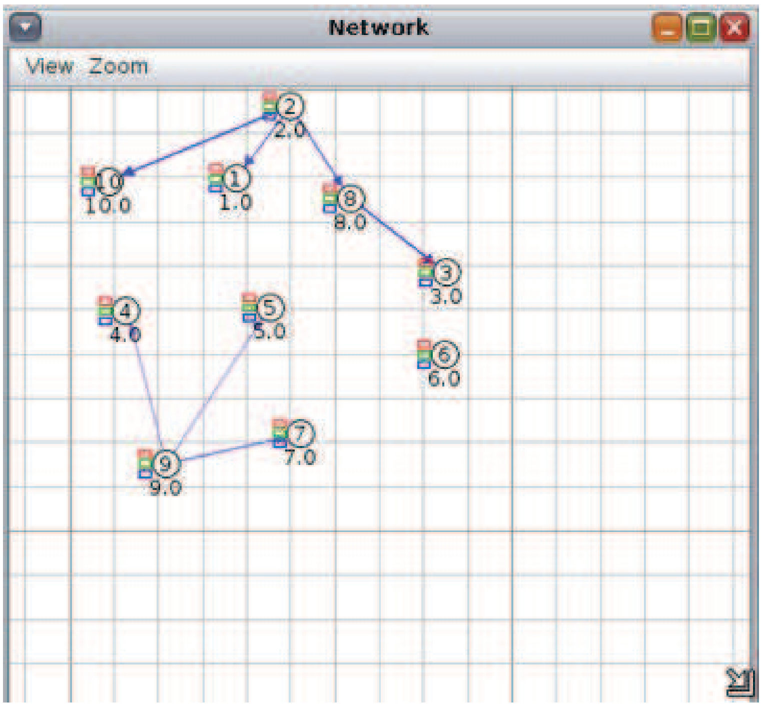


Figure 9: Routing at time instance T1

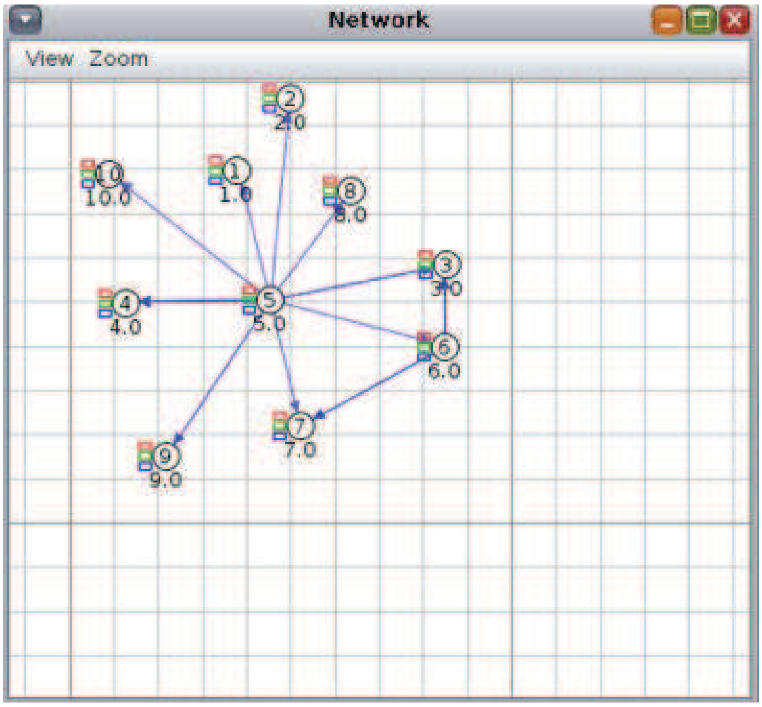


Figure 10: Routing at time instance T2

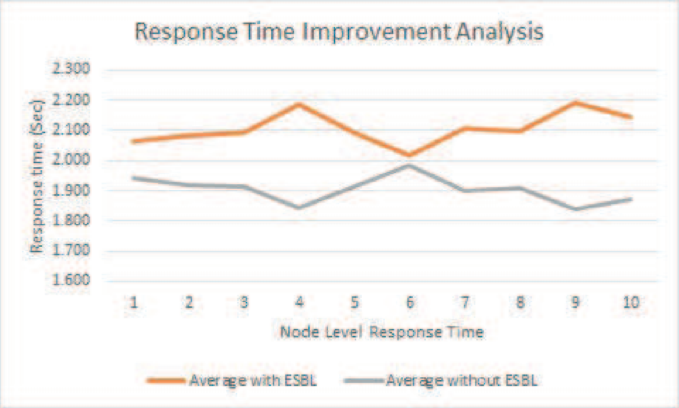


Figure 11: Response time improvement analysis

mising the processing capabilities and energy efficiently.

In the next section of the work, the comparative analysis is carried out.

8 Comparative analysis

To realize the benefits or improvements obtained from any proposed algorithm or framework can be understood by comparing the results with parallel or standard research outcomes. Hence in this section of the work, the comparative analysis is carried out [Tab. 6].

From the comprehension of the parallel research results, it is normal to understand that the vast majority of the heap adjusting calculations have flopped in advocating the IoT organize load adjusting requests as every one of the parameters associated with examining the heap of any virtual machine were given equivalent weight age. In the logical inconsistency, this work proposes a novel weight age metric to legitimize the IoT load adjusting requests as concentrated in the past areas of this work. Considering the constrained power and handling capacities of any IoT hub in the system, this calculation improves the vitality effectiveness and the preparing abilities to a more noteworthy expand. Likewise, the stacking adjusting process time multifaceted nature is decreased altogether.

Table 6: Comparative analysis

Name	Selection Policy	Allocation Policy	Comparison ESLB		
			Energy consumption	Number of node shutdowns	Execution Time
IQR MMT	Minimum Migration Time	Inter Quartile Range	Improved	Improved	Improved
IQR MU	Minimum Utilization	Inter Quartile Range	Improved	Improved	Improved
LR MMT	Minimum Migration Time	Local Regression	Not Improved	Improved	Improved
LR MU	Minimum Utilization	Local Regression	Not Improved	Improved	Not Improved
MAD MMT	Minimum Migration Time	Median Absolute Deviation	Improved	Improved	Improved
MAD MU	Minimum Utilization	Median Absolute Deviation	Improved	Improved	Improved
THR MMT	Minimum Migration Time	Static Threshold	Improved	Improved	Not Improved
THR MU	Minimum Utilization	Static Threshold	Improved	Improved	Improved

Hence, it is natural to realize that the proposed algorithm has outperformed most of the parallel research outcomes.

9 Conclusions

The increasing demand for higher complex data processing applications for managing IoT network data make motivated the application and network designer to push the application on to the cloud computing environment. Considering less dependencies of services and platform from the cloud service models, the major need is consisting of only infrastructure from the cloud service type offerings. With this decision of utilizing the benefits of infra as a service, the demand for balancing the loads on cloud computing is also unavoidable. Observing the parallel standard load balancing strategies, this work identifies that the existing algorithms are not designed to match the demands of IoT networks. Hence this work proposes a novel algorithm for balancing loads for cloud integrated IoT frameworks utilizing the characteristics of loads such as CPU utilization, memory utilization, energy consumption, SLA non-violation and equal distribution of the virtual machines with equal distributions of application loads on virtual machines. The algorithm demonstrates significant improvements over standard algorithms and also improves the IoT network response time by 60.

Bibliography

[1] Ahanger, T.A. (2018). Defense Scheme to Protect IoT from Cyber Attacks using AI Principles, *International Journal of Computers Communications & Control*, 13(6), 915–926, 2018.

[2] Alvi, S. A.; Shah, G. A.; Mahmood, W. (2015). Energy efficient green routing protocol for internet of multimedia things, *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, IEEE, 1–6, 2015.

[3] Balakrishna, G.; Rao, M. N. (2019). Study report on using iot agriculture farm monitoring. *Innovations in Computer Science and Engineering*, Springer, 483–491, 2019.

[4] Di Marco, P.; Athanasiou, G.; Mekikis, P.-V.; Fischione, C. (2016). Mac-aware routing metrics for the internet of things, *Computer Communications* 74, 77–86, 2016.

[5] Evans, D. (2011). The internet of things: How the next evolution of the internet is changing everything, *CISCO white paper* 1–11, 2011.

[6] Guo, J.; Orlik, P.; Zhang, J.; Ishibashi, K. (2014). Reliable routing in large scale wireless sensor networks, *14 Sixth International Conference on Ubiquitous and Future Networks*,IEEE, 99–104, 2014.

[7] Karthikeya, S. A.; Vijeth, J.; Murthy, C.S.R. (2016). Leveraging solution-specifc gateways for cost-effective and fault-tolerant iot networking, *IEEE Wireless Communications and Networking Conference*, 1–6, 2016.

[8] Kim, H.-Y. (2015). An effective load balancing scheme maximizes the lifetime in wireless sensor networks, *5th International Conference on IT Convergence and Security (IC-ITCS)* 1–3, 2015.

[9] Le, Q.; Ngo-Quynh, T.; Magedanz, T. (2014). Rpl-based multipath routing protocols for internet of things on wireless sensor networks, *International Conference on Advanced Technologies for Communications (ATC 2014)*, 424–429, 2014.

[10] Li, K. (2019). Computation Offloading Strategy Optimization with Multiple Heterogeneous Servers in Mobile Edge Computing, *IEEE Transactions on Sustainable Computing*, 1-1, 2019.

[11] Li, L.; Zhou, H.; Xiong, S. X.; Yang, J.; Mao, Y. (2019). Compound model of task arrivals and load-aware offloading for vehicular mobile edge computing networks, *IEEE Access*, 7, 26631–26640, 2019.

[12] Mainetti, L.; Patrono, L.; Vilei, A. (2011). Evolution of wireless sensor networks towards the internet of things: A survey, *SoftCOM 2011, 19th international conference on software, telecommunications and computer networks*,IEEE, 1–6, 2011.

[13] Menouer, T.; Cerin, C. (2017). Scheduling and resource management allocation system combined with an economic model, *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications*, 807–813, 2017.

[14] Oteafy, S. M.; Al-Turjman, F. M.; Hassanein, H. S. (2012). Pruned adaptive routing in the heterogeneous internet of things, *In 2012 IEEE Global Communications Conference(GLOBECOM)*, IEEE, 214–219, 2012.

[15] Oteafy, S. M. and Hassanein, H. S. Towards a global iot: Resource re-utilization in wsns. *In 2012 international conference on computing, networking and communications (ICNC), IEEE* pages 617–622.

[16] Peinl, R.; Holzschuher, F.; Pfitzer, F. (2016). Docker cluster management for the cloud-survey results and own solution, *Journal of Grid Computing*, 265–282, 2016.

[17] Petrioli, C.; Nati, M.; Casari, P. et al. (2013). Alba-r: Load-balancing geographic routing around connectivity holes in wireless sensor networks, *IEEE Transactions on Parallel and Distributed Systems*, 25(3), 529–539, 2013.

[18] Tarek Menouer, C.C.; Leclercq, E. (2018). New multi-objectives scheduling strategies in docker swarmkit. *In International Conference on Algorithms and Architectures for Parallel Processing*, Springer, 103–117, 2018.

[19] Wu, M.; Lu, T.-J.; Ling, F.-Y. et al. (2010). Research on the architecture of internet of things, *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, IEEE, 5, 484–487, 2010.

[20] Xu, Y.; Yin, F.; Xu, W. et al. (2019). Wireless Traffic Prediction with Scalable Gaussian Process: Framework, Algorithms, and Verification, *IEEE Journal on Selected Areas in Communications*, 37(6), 1291–1306, 2019.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 475-488, August 2019.

Facial Expression Decoding based on fMRI Brain Signal

B. Cao, Y. Liang, S. Yoshida, R. Guan

Benchun Cao
1. Zhuhai Laboratory of Key Laboratory of Symbol Computation and Knowledge
Engineering of Ministry of Education, Zhuhai College of Jilin University
Zhuhai, 519041, China
2. Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of
Education, College of Computer Science and Technology,
Jilin University, Changchun, 130012, China
caobc15@mails.jlu.edu.cn

Yanchun Liang
1. Zhuhai Laboratory of Key Laboratory of Symbol Computation and Knowledge
Engineering of Ministry of Education, Zhuhai College of Jilin University
Zhuhai, 519041, China
2. Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of
Education, College of Computer Science and Technology,
Jilin University, Changchun, 130012, China
ycliang@jlu.edu.cn

Shinichi Yoshida
School of Information, Kochi University of Technology,
Kochi 782-8502, Japan
yoshida.shinichi@kochi-tech.ac.jp

Renchu Guan*
1. Zhuhai Laboratory of Key Laboratory of Symbol Computation and Knowledge
Engineering of Ministry of Education, Zhuhai College of Jilin University
Zhuhai, 519041, China
2. Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of
Education, College of Computer Science and Technology,
Jilin University, Changchun, 130012, China
*Corresponding author: guanrenchu@jlu.edu.cn

Abstract: The analysis of facial expressions is a hot topic in brain-computer interface research. To determine the facial expressions of the subjects under the corresponding stimulation, we analyze the fMRI images acquired by the Magnetic Resonance. There are six kinds of facial expressions: "anger", "disgust", "sadness", "happiness", "joy" and "surprise". We demonstrate that brain decoding is achievable through the parsing of two facial expressions ("anger" and "joy"). Support vector machine and extreme learning machine are selected to classify these expressions based on time series features. Experimental results show that the classification performance of the extreme learning machine algorithm is better than support vector machine. Among the eight participants in the trials, the classification accuracy of three subjects reached 70-80%, and the remaining five subjects also achieved accuracy of 50-60%. Therefore, we can conclude that the brain decoding can be used to help analyzing human facial expressions.

Keywords: Brain-computer interface, machine learning, extreme learning machine, fMRI, image processing.

1 Introduction

Brain is the control center of the entire body. It is of great importance and is an indispensable part of us. The brain-computer interface as one of the state-of-the-art issues in the field of science has attracted more and more attentions. If we can judge people's specific behaviors by analyzing the brain, we can understand the brain better [11]. Functional magnetic resonance imaging (fMRI) is a popular technique for studying brain in recent years [11, 18]. It can scan the brain to obtain images and shows the area of the voxel that is activated in the brain when the subject is stimulated by outside stimulus. Through the voxels we can determine which behaviors that the subject was doing are activated in the brain. Medical image processing is a hot issue in the image recognition field. More and more researchers have already invested into the study of medical image processing [12]. Using computers to help people identify medical images can reduce people's errors caused by overwork and improve the accuracy of judgment [12].

Recently, the problem of brain decoding has attracted broad attention. The facial expressions decoding is the main research direction for brain decoding. In daily human interactions, facial expression is one of the most intuitive feelings for humans to communicate. It is crucial for us to understand others' emotions. If the brain activity corresponding to the facial expression can be retrieved, this information can be used to analyze facial expressions. This can be achieved by Brain Computer Interface (BCI), which is a technique to obtain the internal signal of the brain through an external device connected to the brain [16]. At present, electroencephalogram (EEG) and functional magnetic resonance imaging (fMRI) have successfully completed the acquisition of non-invasive signals of brain-computer interface systems [14]. The imaging principle of fMRI is through the magnetic field inside of the machine to affects the activity of the human neuron. The activity of the neuron changes the blood flow and blood oxygen, and fMRI is imaged by measuring these changes. Because of its non-invasive and reproducible features, fMRI is a hot topic for medical researchers and experts. Studies [7] have pointed that fMRI not only can interpret simple movement instructions, but also can interpret complex advanced thinking activities that are related to certain regions of the brain. Therefore, fMRI technology is more suitable for decoding human facial expressions [6]. In particular, the research on the location of brain regions has been widely used.

The brain information decoding technology refers to using fMRI and machine learning algorithms to estimate the subject's facial expressions under the stimulation of expression images according to the acquired brain signals. However, low recognition accuracy is a major problem in brain information decoding, which has limited this technology's fast spreading. The main task of this paper is to extract the brain image features that have been stimulated to show different states through the fMRI brain images analysis with higher classification accuracy. Comparing with the time series, we find out different stimulus labels corresponding to different brain image features.

The remainder of this paper is organized as follows. Section 2 introduces the fundamental concepts of related algorithms. Section 3 elaborates the data parsing process of fMRI data. Section 4 presents the specific implementation process of the experiment and the analysis of related experimental results. Finally, the conclusion is drawn in Section 5.

2 Related theory

To better understand our model, this section gives a detailed introduction and explanation about all the related algorithms including cross validation, support vector machine algorithm, extreme learning machine algorithm and brain decoding. For each algorithm, this section specifies its principles and implementation steps.

2.1 Cross validation algorithm

Cross-validation is an algorithm used to evaluate the performance of classifiers [1]. Its main idea is to divide data sets into training sets and verification sets. Training sets are mainly used to train classifiers and verification sets are used to verify the classification performance. According to the data partitioning, cross-validation methods are divided into three categories hold-out method [2], k-fold cross validation method [15] and leave-one-out cross validation method [10]. The k-fold cross-validation method divides the original data set into k parts, which is divided into k parts. Each of the k parts is used as verification set; the other k-1 parts are used as the training set. We get k models and take the arithmetic average as the accuracy of the model. The advantage of the k-fold cross-validation method is that the segmentation is reasonable so that the data set can be fully utilized, the probability of occurrence of the under-fitting problem is reduced and the training data is greatly increased, and the trained model can be more optimized. Because k-fold cross-validation can effectively avoid over learning and less learning and the result is convincing, we used k-fold cross-validation method.

2.2 Support Vector Machine

Support Vector Machine (SVM) is a very popular machine learning algorithm [17]. The original support vector machine was specifically designed to solve the linear divisibility problem. Introducing kernel functions, SVM can map low-dimensional data to high dimensions and solve the linear indivisibility problems. The purpose of SVM is to generate a hyperplane that divides the linearly separable data into two classes and make the point closest to this line as far away as possible from the hyperplane. The interval which is called the geometric interval should be as large as possible so that the data can be completely separated. We assume that the formula for the linear segmentation plane is $\mathbf{w}^{\mathrm{T}} + b = 0$.

The geometric interval can be obtained by solving the following optimization problem:

$$f(x) = Max_{w,b}\frac{1}{2}\|w\|^2. \tag{1}$$

Subject $to : y^i(wx^i + b) \geq 1$

where the weight vector w and the offset value b represent the parameters of the classification plane. For nonlinear classification problems, SVM cannot classify them in the low-dimensional space and the kernel function solves this problem well. The kernel function can map the data from the low dimension to high dimension. With kernel functions, SVM can be used to classify non-linear problems. There are different types of kernel functions such as sigmoid kernel, Gaussian kernel and tanh kernel.

2.3 Extreme Learning Machine

Extreme Learning Machine (ELM) is a neural network with a single hidden layer which randomly initializes weights and do not need to adjust parameters during training [9]. The ELM consists of three layers: the input layer, hidden layer, and output layer. The learning speed of ELM is fast because ELM has only one single hidden layer. ELM initializes weights randomly and there is no need to adjust offsets and weights in the training process.

Figure 1 shows a neural network with single hidden layer. If there are N groups input data for the neural network, it can be expressed as

$$(x_j, t_j) \tag{2}$$

where

$$x_j = [\mathbf{x_{j1}}, \mathbf{x_{j2}}, \cdots, \mathbf{x_{jn}}]^{\mathrm{T}} \in R^n$$

$$t_j = [\mathbf{t_{j1}}, \mathbf{t_{j2}}, \cdots, \mathbf{t_{jm}}]^{\mathrm{T}} \in R^m$$

we assume that the number of hidden nodes in the neural network is L.

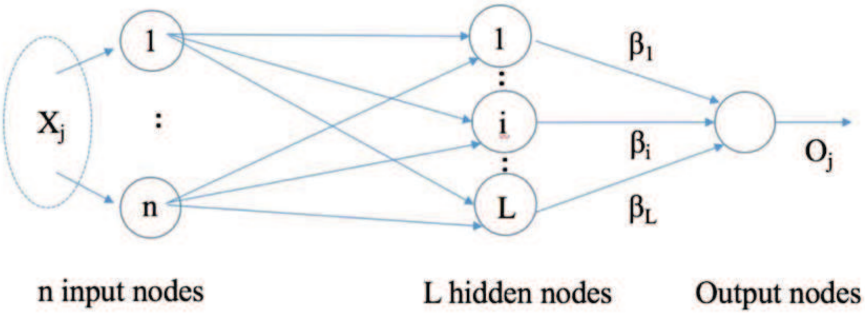


Figure 1: Single hidden layer neural network

Then the neural network can be expressed as

$$\sum_{i=1}^{L} \beta_i g(\omega_i x_j + b_i) = o_j, j = 1, 2, ..., N. \tag{3}$$

where $g(x)$ denotes the activation function of the hidden layer node,
$\omega_i = [\omega_{\mathbf{i1}}, \omega_{\mathbf{i2}}, \cdots, \omega_{\mathbf{in}}]^{\mathrm{T}}$ denotes the input weight of the i-th node on the hidden layer, and b_i denotes the offset of the i-th node on the hidden layer,
$\beta_i = [\beta_{\mathbf{i1}}, \beta_{\mathbf{i2}}, \cdots, \beta_{\mathbf{im}}]^{\mathrm{T}}$ represents the output weight of the i-th node of the hidden layer,
$\omega_i \cdot \omega_j$ indicates the inner product of the weights ω_i and ω_j.

The goal of training single hidden layer neural network is to minimize the error between the output and the expected value of the network output layer $min(\sum_{j=1}^{N} \|o_j - t_j\|)$.

Hereinto,

$$\sum_{i=1}^{L} \beta_i g(\omega_i x_j + b_i) = t_j, j = 1, 2, ..., N.$$

It can be expressed in the form of matrix

$$\mathbf{H} \cdot \beta = \mathbf{T} \tag{4}$$

where $\mathbf{H}$ represents the output matrix of the hidden layer node,
β represents the weight of the output of the hidden layer,
$\mathbf{T}$ represents the target output of the output layer:

$$\mathbf{H}(\omega_1, ..., \omega_L, b_1, ..., b_L, x_1, ..., x_N) = \begin{bmatrix} g(\omega_1 \cdot x_1 + b_1) & \cdots & g(\omega_L \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\omega_1 \cdot x_N + b_1) & \cdots & g(\omega_L \cdot x_N + b_L) \end{bmatrix} \tag{5}$$

$$\beta = [\beta_{\mathbf{1}} \cdots \beta_{\mathbf{L}}]^{\mathrm{T}} \tag{6}$$

$$\mathbf{T} = [\mathbf{t_1} \cdots \mathbf{t_N}]^{\mathrm{T}}. \tag{7}$$

In order to optimize the neural network, we expect to obtain $\widehat{w_l}, \widehat{b_l}$ and $\widehat{\beta_l}$, making

$$\|\mathbf{H}(\widehat{w_l}, \widehat{b_l}) \cdot \beta - \mathbf{T}\| = min_{w,b,\beta}\|\mathbf{H}(\widehat{w_i}, \widehat{b_i}) \cdot \beta - \mathbf{T}\| \tag{8}$$

where $i = 1, 2, \cdots, L$.

This is equivalent to minimizing the loss function:

$$\mathbf{E} = \sum_{j=1}^{N} \| \sum_{i=1}^{L} \beta_i g(\omega_i x_j + b_i) - t_j \|^2 \tag{9}$$

For solving this kind of problem, the gradient descent method is often used to find the optimal solution. However, using the gradient descent algorithm, we need to adjust the parameters of all hidden layers based on expert experience or optimization algorithms. To avoid this problem, we introduce extreme learning machine (ELM).

For ELM, it is no need to adjust the parameters and the offsets as long as the weights are initialized. The training process can be summarized as: $\mathbf{H} \cdot \beta = \mathbf{T}$, and we can get output weight $\widehat{\beta}$

$$\widehat{\beta} = \mathbf{H}^{+} \cdot \mathbf{T} \tag{10}$$

where $\mathbf{H}^{+}$ represents the generalized inverse matrix of matrix $\mathbf{H}$. Huang et al. have proved that the norm of the solution $\widehat{\beta}$ is not only smallest but unique [8].

2.4 Brain decoding

The flowchart of brain information decoding is shown in Figure 2. First, by giving stimulus images and measuring brain activity data, the facial expressions of the subjects are recorded according to the time series. Then, we extract the feature vectors under this stimulus according to activated voxels. Finally, the obtained feature vectors are corresponding to the facial expressions through the time series to form a feature vector and label input classifier for counting the recognition accuracy.

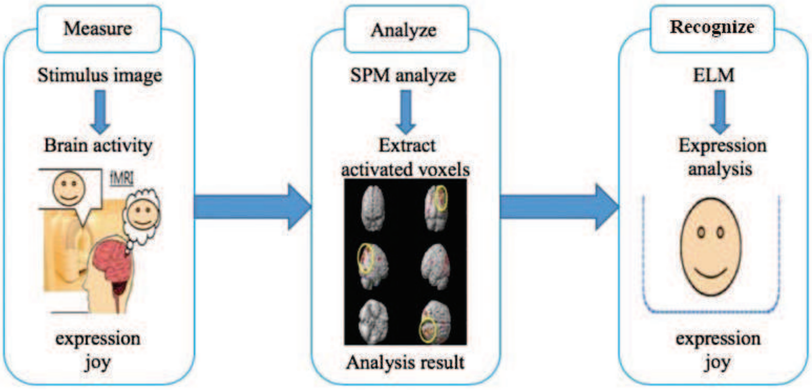


Figure 2: Brain information decoding processes

3 fMRI data analysis

For the preprocessing procedure, we use Statistical Parametric Mapping (SPM) [13] to analyze the obtained fMRI images, personal parsing process and group parsing process. The process of fMRI Data Analysis is shown in Figure 3.

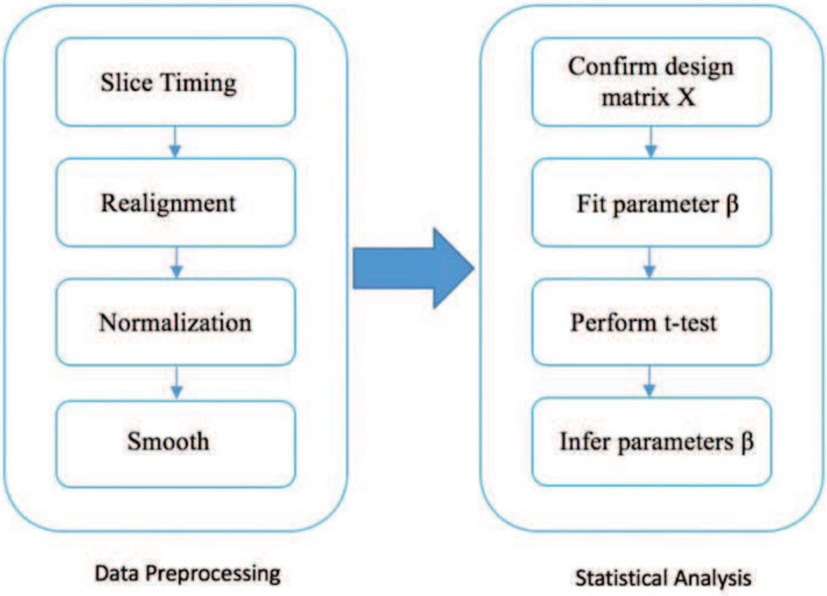


Figure 3: fMRI data analysis process

3.1 Data preprocessing

Before analyzing the data of the brain image, we need to preprocess the brain data and use the SPM software to complete the process. It mainly consists of four steps: Slice Timing, Realignment, Normalize, and Smooth.

Slice Timing. During the collection of brain images, because we cannot scan the entire brain image at a time, we scan brain in slice. In the process of scanning, there will be slight differences in the acquisition time of each slice; therefore we need to correct time difference in the acquisition slice to ensure that the acquisition time is the same for each slice. Slice timing is used to correct the difference in time between the slice and the slice in the brain image.

Realignment. Although we fix the head of the subjects, some subjects still had slight head shake during the experiment. The process of realignment is to unify all the head portraits of the subjects in the entire test process, and the purpose is to correct head shake. Assuming that subject's brain shaking category is within an acceptable category, then we correct it by some methods to make it close to the exact value, but if it is not in this category, we need to delete the portrait. In general, the range of head movements we examined was that the translation does not exceed 2.0 mm and the rotation does not exceed 2.0 degrees.

Normalization. Because different subjects have different brain sizes and shapes, we standardize the brain images of the subjects before carrying out the experiment in order to deal with them in a unified way. We place the brain images of the subjects with differences in a common space and use the same norms to describe the specific sites. For example, when there are multiple experimental participants with different head shapes, if we want to make an experiment for them to select an area of the brain, we should develop a standard spatial brain template to achieve accurate positioning.

Smooth. Since fMRI contains a large amount of noise, to minimize the impact of noise and obtain images with less difference from the original data, we smooth the acquired image. In SPM software, smoothing is the process of deconvoluting the acquired brain image using a Gaussian function.

3.2 Statistical analysis process

Directly using the preprocessed data to data analysis will bring a lot of unnecessary problems. To enable the entire experiment to be proceed smoothly, we divide the preprocessed data into two areas, the interest region and other regions. In data analysis process, we extract feature vectors directly from the regions of interest (ROI) [4]. In the statistical analysis process, a very important statistical analysis model is used, namely the generalized linear model (GLM) [19].

GLM is an extension of the linear model. The independent variables should be continuous values rather than discrete integers or data, and the relationship between the expected values of the independent and dependent variables are linear. The independent variables of the generalized linear model are not only limited to continuous numerical data, but discrete integers are also satisfactory. The other difference is that we need to determine the connection function when modeling GLM.

In simple terms, GLM is based on the assumption that the test result (represented by Y) at each pixel is a linear combination of certain parameters X. These parameters are not only related to the specific brain regions, but are related to the experimental task and the experimental time. The matrix composed of these parameters is called the design matrix. It can be expressed as $\mathrm{Y} = \beta\mathrm{X} + \varepsilon$ where ε represents the error, we have changed the parameters of the solution after the transformation through the GLM. We originally asked for a statistical analysis of the variable Y, and now we are solving the parameter β.

We can get brain activation map by fitting statistics to β. The statistical inference procedure for parameter β is as follows:

Step 1: Confirm the accuracy of design matrix X.

Step 2: The least squares method is used to fit the parameter β, and make the error sum $\sum_{i=1}^{N} \varepsilon^2$ reach the minimum.

Step 3: Perform t-test or F-test on parameter β.

Step 4: According to t test or F test statistical results to infer parameters β.

4 Experiment and discussion

In our experiment, the fMRI image of the subject under stimulation was first obtained by Magnetic Resonance. Then the fMRI image analysis software SPM was used to analyze the fMRI images of different subjects stimulated by different expression images (anger and joy) respectively. The activated brain voxel area was obtained under the corresponding stimulus. This experiment allows participants to maintain a consistent facial expression and the stimulus images they see. The voxel area that is activated by the time series corresponds to the stimuli portrait. The obtained active voxel area was used as the feature vector and the facial expressions of the subjects were used as labels, which are input into the classifier to evaluate the performance of the algorithm.

Paul Ekman claims that there are six basic emotions in human emotion [5]. They are "anger", "disgust", "sadness", "happiness", "joy", and "surprise". In our case, anger and joy will be used to predict.

4.1 Experimental data

There are 8 subjects participated in the experiment, where 5 were males and 3 were females. All the subjects were from 21 to 22 years old and in good health to avoid the effects of gender and age on the experimental results. The safety of the fMRI equipment was explained to the subjects before the experiment and a confidentiality agreement on the experimental results was signed. The stimuli used in this study are anger and joy images. To emphasize facial features, the clothing and background of the stimuli are filled with gray to reduce the impact on the experimental results.

In the experimental design, 216 seconds brain activity was imaged for each subject. Eight brain scans were performed for each subject and 72 brain activity slices were obtained each time during this process. To reduce bold signal, we show 3 black background scans between the stimuli to stabilize the brain. The stimulating images randomly presented 64 facial portraits, including 32 "joy" facial expressions and 32 "angry" facial expressions. The specific form is shown in Figure 4.

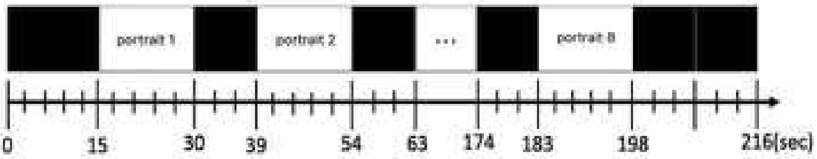


Figure 4: Brain scan time

This following step is mainly carried out through three steps. First, different facial expression images were randomly presented to stimulate the subjects, and the subjects were asked to make corresponding facial expressions with the images, mainly including two expression images of anger and joy. Then the brain data of subjects was scanned by MRI to images. The obtained brain images are analyzed using SPM to obtain useful information. Finally, the data obtained by the experiment was input to the classifier.

4.2 Data preprocessing and analysis

The original image taken from the fMRI device is a DICOM format image. To use SPM for data analysis, we need to convert the image to the desired analysis format. This experiment uses the MRIConvert software to convert the image format. In the process of analyzing the brain images of the subjects, we used SPM software to identify the brain activation voxels. The personal analysis process is as follows: 1. Input preprocessed brain activity data into generalized linear model (GLM). 2. Use the design matrix created to calculate the regression coefficient β of the GLM model 3. Perform T-test on the subject's brain image to determine which brain voxels are activated 4. Analyze the brain regions of activated voxels belongs from the display image (reference Brodmann region system [20]).

When the stimulus image is joy or anger, the analysis results for each subject's fMRI image are presented in Table 1 and Table 2, respectively. The pulse repetition time (TR) of the subjects A, B, C, D was 3090ms, and E, F, G, H was 3480ms. The pulse repetition time refers to the interval where the brain is scanned using an MRI.

The group analysis is to analyze the information collected by all subjects as a whole. In the group analysis, the contrast file obtained by personal analysis of the subjects was used to analyze the whole group of subjects. The results of eight subjects group analysis are shown in Figure 5. When the stimulus image is "joy", the activated voxels are located in the lower parietal regions

Table 1: Brain image analysis results on the stimulus of joy

Subject	P value	T value	Activated Broadmann region
A	0.01	2.89	17-19,37,39
B	0.001	4.05	17-19,7
C	0.001	5.11	9-19,23-40,44-47
D	0.001	4.15	9-12
E	0.001	4.82	9-12
F	0.0001	9.51	17-19,37
G	0.001	3.70	9,46
H	0.001	4.00	9-12,23-27

Table 2: Brain image analysis results on the stimulus of anger

Subject	P value	T value	Activated Broadmann region
A	0.01	3.02	9-12,17-19
B	0.001	4.11	17-19
C	0.001	4.04	37,20,21
D	0.001	4.34	7-16,20-22,39-47
E	0.001	5.92	9-12
F	0.0001	3.72	Lateral geniculate body, posterolateral nucleus
G	0.001	3.59	9,11-16.25-33,34-36,46,47
H	0.001	4.02	9-12,45-47

of the 40 and 41 regions of the Brodmann region, mainly related to perception, vision, reading and language-functional area.

When the stimulus image is "angry", the significance level is set to $p < 0.01$. At this time, the activated voxels are located in the inner and outer occipital regions of the 19 region Brodmann area. This area is mainly visual, color vision, and movement–related areas. It can be seen from the Figure 5 that amygdala is not completely activated. According to the relevant analysis, as the amygdala is the brain part on the facial expression recognition [3] , the amygdala of the test was only activated a little. Furthermore, we can see that the visual and other areas are more fully activated.

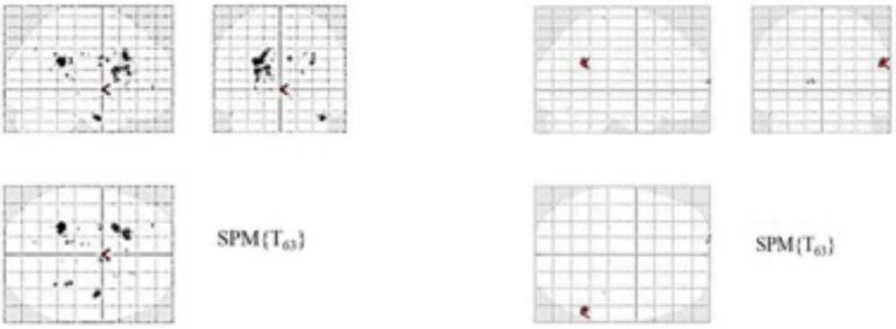


Figure 5: Brain activity state (joy, anger) of group analysis

4.3 Classification results comparison

According to the above discussion, we have already analyzed the fMRI image of the subject through SPM software and obtained the feature vector. According to the time series record in the experiment, we can associate the feature vector with the facial expression. Thus, a data containing feature vector and feature label is generated. To achieve high recognition accuracy, the selection of the classifiers is of great importance.

We classify the obtained data using SVM and ELM, respectively. In this comparison experiment, the number of ELM hidden layer nodes is 1000 nodes and the Sigmoid function is the activation function. And the C value of SVM is 0.5 and the Sigmoid function is selected as the kernel function, we use the grid search algorithm to find the optimal super parameters for the sigmoid function $K(x, y) = tanh(\gamma \cdot x^t \cdot y + g)$ in order to get the optimized SVM model. In the parameter selection, we respectively let $\gamma = (1, 2, 3, 4)$ and $g = (0.2, 0.4, 0.6, 0.8)$. According to the experimental results, we get that the function is optimal when $\gamma = 2andg = 0.6$. From the results it can be found that no matter what the data obtained in the stimulation of the joy images or anger images, the classification result of the ELM is better than SVM. Figure 6 shows the comparison of the classification accuracy under the stimulation of the joy images. Figure 7 shows the comparison of the classification accuracy under the stimulation of the anger image. Therefore, we use ELM to classify the obtained fMRI images.

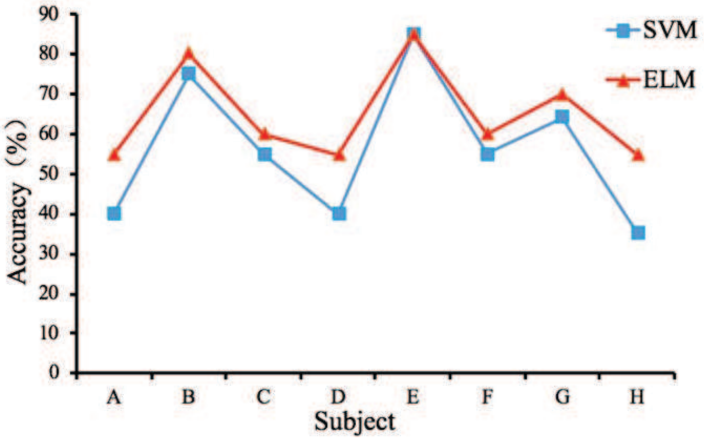


Figure 6: Performance comparison of SVM and ELM for joy images stimulation

Because the number of hidden layer nodes and the selection of activation function in the ELM directly affect the classification performance of the entire ELM, To pursue the highest accuracy, we need to determine the optimize number of hidden layer nodes and activation function.

Figure 8 shows the effect of the active functions and the number of hidden nodes on the classification accuracy of ELM for the joy images. From Figure 8 we can see that the overall classification performance of the Sigmoid function is higher than the tanh function. Although using 2500 hidden nodes, the former function is not good as tanh, it does not affect the overall trend. Therefore, we use the Sigmoid function as the activation function of the classifier ELM. From the curve of the Sigmoid function, we can see that the number of nodes in the hidden layer ranges from 500 to 4000 and every 500 nodes are located in one interval. We can see that the classification performance reaches the best when the number of nodes is 1000. Therefore, the number of hidden layer nodes of the ELM classifier is chosen as 1000 nodes. In summary,

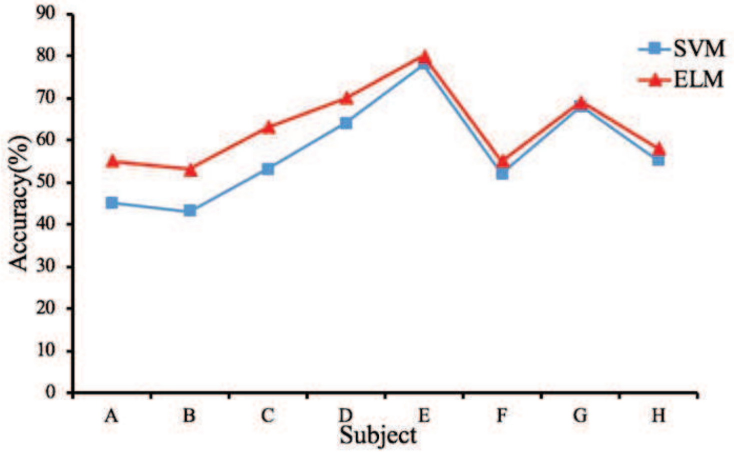


Figure 7: Performance comparison of SVM and ELM for anger images stimulation

the amount of hidden layer nodes is 1000 and the Sigmoid function is selected as the activation function.

4.4 Discussion

We use cross-validation to evaluate the classification performance of ELM. It not only enables ELM to make more effective predictions, but also reduces the chance of overfitting. The k fold cross-validation is used to evaluate the performance of the ELM. In our experiment, let k be 10, that is, we divide the data into 10 groups and use 9 groups each time as the training set to train the ELM, the other group as test set. The average accuracy of the ten groups of classification is the classification accuracy of the classifier. In this experiment, the classification accuracy of each subject is shown in Table 3.

Table 3: The accuracy of ELM on each subject data

Subject	**Classification accuracy(joy)**	**Classification accuracy (anger)**
A	55%	55%
B	80%	55%
C	60%	62%
D	53%	70%
E	88%	80%
F	60%	58%
G	68%	70%
H	55%	60%

Table 3 lists the classification accuracy of the brain images acquired by fMRI equipment from the 8 subjects under the effect of cross-validation and ELM classifiers. It can be seen that the accuracy of 3 subjects (subjects B, E, G) reached 70-80% and the remaining 5 subjects reached 50-60%. According to these results, it can be found that facial expressions can be analyzed by analyzing fMRI brain images. Among these subject's results, the accuracy of subject A is low, and the reason for the low accuracy rate may be due to noise. It may also be caused by the

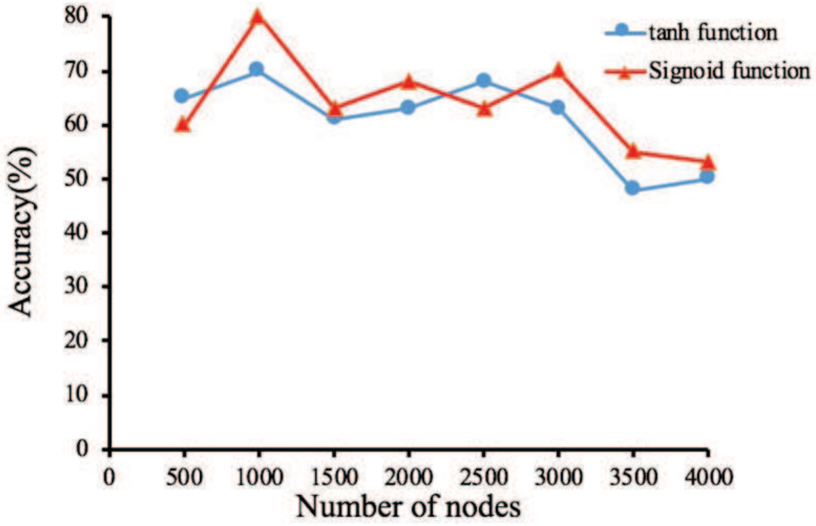


Figure 8: Effect of the activation functions and the amount of hidden layer nodes

insensitivity of the subject to the stimulating image.

5 Conclusion

In this paper, the machine learning algorithm ELM is applied to the classification of fMRI data. From the results of personal analysis, we can see that the accuracy of the recognition of 3 subjects (B, E, and G) in the 8 subjects reached 70-80%, and the accuracy of the remaining 5 subjects also reached 50-60%. It can be seen that using the brain image obtained by fMRI we can analyze people's facial expressions. From the results we can see that the activated voxels are usually located in the brain regions with visual, ideological, and cognitive functions. This article also provides the proof that we can analyze the facial expressions of the subjects by analyzing the fMRI images. It should be pointed out that in the near future, the stimulatory images may not be limited to still images. It is possible to study by using video to stimulate subjects and then observe the experimental results.

6 Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (61572228, 61602207), the Jilin Provincial Key Laboratory of Big Data Intelligent Computing (20180622002JC), the Key Technological Research Projects in Jilin Province under Grant 20190302107GX, the Special Research and Development of Industrial Technology of Jilin Province under Grant 2019C053-7, Guangdong Premier Key-Discipline Enhancement Scheme (2016GDYSZDXK036), and Guangdong Key-Project for Applied Fundamental Research (2018KZDXM076).

Bibliography

[1] Arlot, S.; Celisse, A.(2010). A survey of cross-validation procedures for model selection, *Statistics Survey*, 4, 40-79, 2010.

[2] Blum, A.; Kalai, A.(1999). Beating the hold-out:bounds for K-fold and progressive cross-validation, *Proceedings of the twelfth annual conference on Computational learning theory. ACM*, 203-208, 1999.

[3] Davis, M; Whalen, P. J.(2001). The amygdala: vigilance and emotion, *Mol Psychiatry*, 6(1), 13-34, 2001.

[4] Desikan, R.S.; Segonne, F; Fischl, B.(2006). An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest, *Neuroimage*, 31(3), 968-980, 2006.

[5] Ekman, P.(1992). An argument for basic emotions, *Cognition* & *Emotion*, 6(3-4), 169-200, 1992.

[6] Forman, S D; Cohen, J.D.; Fitzgerald, M. et al.(1992). Improved Assessment of Significant Activation in Functional Magnetic Resonance Imaging (fMRI): Use of a Cluster-Size Threshold, *Magnetic Resonance in Medicine*, 3(5), 636-647, 2010.

[7] Gilead, M.; Liberman, N.; Maril, A.(2013). The language of future-thought: An fMRI study of embodiment and tense processing, *Neuroimage*, 65(2), 267-279, 2013.

[8] Huang, G.B.; Zhu, Q.Y.; Siew, C.K.(2004). Extreme learning machine: a new learning scheme of feedforward neural networks, *IEEE International Joint Conference on Neural Networks*, 2, 985-990, 2005.

[9] Huang, G. B.; Zhu, Q. Y.; Siew, C.K.(2006). Extreme learning machine: Theory and applications, *Neurocomputing*, 70(1), 489-501, 2006.

[10] Kearns, M; Ron, D.(1997). Algorithmic stability and sanity-check bounds for leave-one-out cross-validation, *Neural Computation*, 11(6), 1427-1453, 1997.

[11] Logothetis, N.K.; Pauls, J.; Augath, M.(2001). Neurophysiological investigation of the basis of the fMRI signal, *Nature*, 412(6843), 150-157, 2001.

[12] Lehmann, T.M; Gonner, C.; Spitzer, K.(1999). Survey: interpolation methods in medical image processing, *IEEE Transactions on Medical Imaging*, 18(11), 1049-1075, 1999.

[13] Litvak, V.; Mattout, J.; Kiebel, S. et al. (2011). EEG and MEG Data Analysis in SPM8, *Computational Intelligence and Neuroscience*, 2011(3), 852961, 2011.

[14] Michel, C.M; Murray, M.M.; Lantz, G.(2004). EEG source imaging, *Clinical Neurophysiology*, 115(10), 2195-2222, 2004.

[15] Rodriguez, J.D.; Perez, A.; Lozano, J.A.(2010). Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation, *IEEE Computer Society*, 32(3), 569–575, 2010.

[16] Schalk, G.; Mcfarland, D.J.; Hinterberger, T.(2004). BCI2000: a general-purpose brain-computer interface (BCI) system, *IEEE Transactions on Bio-medical Engineering*, 51(6), 1034–1043, 2004.

[17] Suykens, J.A.K.(1999). Least squares support vector machine classifiers: a large scale algorithm[C]. *European Conference on Circuit Theory and Design*, 1999, 839-842, 1999.

[18] Tugui, A. (2014). GLM Analysis for fMRI using Connex Array *International Journal of Computers Communications* & *Control*, 9(6), 768–775, 2014.

[19] Ziegel, E.R.(2012). An Introduction to Generalized Linear Models[J]. *Technometrics*, 44(4), 406-407, 2012.

[20] Zilles, K; Amunts, K.(2010). Centenary of Brodmann's map–conception and fate, *Nature Reviews Neuroscience*, 11(2), 139-145, 2010.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 489-502, August 2019.

Application of Improved Collaborative Filtering in the Recommendation of E-commerce Commodities

D. Chang, H.Y. Gui, R. Fan, Z.Z. Fan, J. Tian

Dan Chang
School of Economics and Management
Beijing Jiaotong University, China
No.3 Shang Yuan Cun, Haidian District, Beijing, China
6787@bjtu.edu.cn

Haoyu Gui
School of Economics and Management
Beijing Jiaotong University, China
No.3 Shang Yuan Cun, Haidian District, Beijing, China
17120610@bjtu.edu.cn

Rui Fan*
School of Economics and Management
Beijing Jiaotong University, China
No.3 Shang Yuan Cun, Haidian District, Beijing, China
*Corresponding author: 17120607@bjtu.edu.cn

Zezhou Fan
School of Economics and Management
Beijing Jiaotong University, China
No.3 Shang Yuan Cun, Haidian District, Beijing, China
17125463@bjtu.edu.cn

Ji Tian
Beijing Research Institute of Automation Machinery Industry Co, Ltd
No.1 Jiaochangkou Street, Xicheng District, Beijing, China
15010118013@163.com

Abstract: Problems such as low recommendation precision and efficiency often exist in traditional collaborative filtering because of the huge basic data volume. In order to solve these problems, we proposed a new algorithm which combines collaborative filtering and support vector machine (SVM). Different with traditional collaborative filtering, we used SVM to classify commodities into positive and negative feedbacks. Then we selected the commodities that have positive feedback to calculate the comprehensive grades of marks and comments. After that, we build SVM-based collaborative filtering algorithm. Experiments on Taobao data (a Chinese online shopping website owned by Alibaba) showed that the algorithm has good recommendation precision and recommendation efficiency, thus having certain practical value in the E-commerce industry.

Keywords: recommendation precision, recommendation efficiency, support vector machine (SVM), collaborative filtering.

1 Introduction

With the rapid development of the Internet and mobile Internet, the E-commerce industry is booming with broad attention from all walks of life. According to Research Report on Market Prospect and Invest Opportunity of E-commerce Industry in China from 2018 to 2023, the overall transaction scale of China's E-commerce reached 24.1 trillion Yuan in 2017, an increase of 17.4%. With the gradual improvement of the E-commerce industry, it is estimated that its transaction scale will reach 28.4 trillion Yuan in 2018, a year-on-year increase of 17.8% [32]. The flourish of the E-commerce industry has led to the explosion of different kinds of data. And the data, which contain great value, are invisible assets for the E-commerce industry. However, not all data are valuable, so users have to spend much time in extracting useful and specific information from a vast amount of data.

With the increasingly booming information in the E-commerce industry and the extraction of valuable information, recommender systems emerge as the times demand – E-commerce websites begin to solve the problem of "information overload" [3] through recommender systems. Commodity recommender systems can record user's characteristic information and their behavioral information such as purchasing and browsing. By analyzing the information obtained from modeling of user's preference, commodity resources that fit user's potential demand or may interest them will be extracted from the commodity information on E-commerce websites, and then recommended to users. Collaborative filtering, with many advantages such as no need to consider the content of recommendation item, offering novel recommendation, little disturbance while browsing websites, as well as easy to achieve technically, becomes the basic algorithm of recommender systems.

Traditional collaborative filtering has the limitations of recommending only on the basis of single indicator — either user's marks or comments. However, marks and comments should not be studied separately, because inconsistency between marks and comments often exists in real life. For example, a user may give a high mark for a commodity, but a dissatisfied comment at the same time and people's cognition on mark differs. Therefore, traditional collaborative filtering is defective in terms of precision.

Nowadays, it is a research hotspot to combine two or several different methods to solve practical problems to make up for the defects of traditional algorithm. For example, in [30] Zhao et al. combined the lexicographic method and Pareto method to optimize the flight ground support capability of airport. Also, to extend the classical vehicle routing problem, they proposed a time-dependent and bi-objective vehicle routing problem with time windows [29]. Therefore, we pretend to combine SVM and collaborative filtering recommendation to improve the recommendation efficiency and precision.

Before giving comprehensive grades, commodities are divided into positive-feedback commodities and negative-feedback commodities in the paper with SVM, namely commodities users like and dislike. Collaborative filtering recommendation is only implemented on positive-feedback commodities. The original data for recommendation reduce greatly due to the classification in advance, so the improved collaborative filtering increases efficiency compared with the traditional one. Finally, online data on Taobao are used to verify that the improved collaborative filtering promotes recommendation precision and efficiency significantly.

2 Literature review

Collaborative filtering is most commonly used in recommender system. It was first proposed by scholars such as Goldberg in 1992, and implemented in TaPestry — the experimental mail system, enabling the system to extract user-interested and effective email list [6]. Mainstream collaborative filtering today can be divided into user-based collaborative filtering and item-based collaborative filtering. At present, collaborative filtering is the research focus in the academic circle. As traditional collaborative filtering has drawbacks such as cold start, data sparsity and poor extensibility, in order to come up with better solutions to these problems, scholars now focus on the improvement of traditional collaborative filtering.

For example, in [7] Guo et al. proposed a novel method called "Merge" to incorporate social trust information, and supplement user preference by merging users' trusted neighbor ratings. In [8] Hu et al. integrated time information into collaborative filtering similarity measure in collaborative filtering algorithm, and designed a hybrid personalized random walk algorithm; Yong-ping Du et al. [5] proposed item-based RBM, and used deep and multilayer RBM network structure to solve the problem of data sparsity; Sedhain et al. [20] generalized matrix algebra framework, and they doesn't need the target user's data when the side information is available ; Jian Wei et al. [25] put forward two models on the basis of a framework based on tight-coupling collaborative filtering and the in-depth study into neural network; A. Murat Yagci et al. [26] focused on frequent co-occurrence items and proposed SASCF to eliminate the cold start of the system; Su Hongyi et al. [22] proposed a new algorithm involving time decay factor in the CF algorithm, and deployed time weights on the MapReduce parallel computing framework ; Xiuju Liu et al. [13] presented a new algorithm of CF-ISEGB, and took the influence sets of current e-learning groups into consideration to effectively solve problems caused by sparse data sets.

Besides, recommendation precision and recommendation efficiency are also two important indicators to assess collaborative filtering. Therefore, many scholars have improved the algorithm by promoting its recommendation precision and efficiency. For example, Mehrbakhsh Nilashi et al. [17] provided the probability of precise recommendation by considering users' preference in many aspects of the items, and introduced vague method to eliminate the uncertainty of users' preference. In [18] Mahdi Nasiri et al. promoted the predictive accuracy of collaborative filtering by initialized factor matrix. Feng Zhang et al. [27] designed linear time algorithm to calculate similarity, thus reducing the time of assessment. In [18] Zhongya Wang et al. calculated PB-level data by parallel computing and proposed effective collaborative filtering based on multi-GPU. Kasra Madadipouya in [14] added location factors to the traditional film collaborative filtering, and improved the accuracy and the quality of recommendation in practical application. Nicola Barbieri in [2] proved that basic probability framework is useful in the generation of the recommendation list, and enhanced the accuracy of recommendation. In [10] Gai Li et al. proposed a new model named PPMF by using RankRLS to solve the problem of low recommendation accuracy and the high cost. With improved algorithms above, scholars have promoted recommendation precision and efficiency. Therefore, it is of significance to improve traditional recommendation algorithm from the perspectives of recommendation precision and recommendation efficiency.

The collaborative filtering in the paper adopts the classification model of SVM. As an implementation method of statistical theory in practice, SVM maps input variable on high-dimensional space by nonlinear mapping, then constrains questions through quadratic optimization and finds out the optimal classification hyperplane, thus maximizing the distance between data and the optimal classification hyperplane. The core concept is to reduce the error of classification to the greatest extent. Many scholars at home and abroad use SVM classifier to classify some data and establish prediction model. Huayu Li et al. separated data into positive and negative feedbacks by a SVM-like task [11]. Uricar et al. in [23] made classification with SVM after figuring out deep characteristic data represented by people's facial expressions, thus predicting their age, gender and smiles. In [15] Asha S. Manek et al. combined feature extraction with SVM and made sentimental classification among online film reviews to predict the popularity of a film. Anish Jindal et al. in [9] combined DT with SVM to precisely predict electricity theft, reducing false alarms greatly. A.S. Ahmad et al. in [1] used the Least Square Support Vector Machine (LSSVM) to forecast electrical energy consumption of buildings. In [21] Selakov et al. used the combination of PSO and SVM to forecast short-term electrical load according to the significant temperature variations. Dongwen Zhang et al. in [25] used SVM in sentimental classification to extract the valuable information.

As SVM can do well in predicting data, scholars at home and abroad combine it with collaborative filtering to predict items or products users may like and then recommend them to users. In order to solve the problem that recommender system is easy to be attacked by shilling, Wei Zhou et al. in [31] combined SVM with TIA and used borderline SMOTE to relieve class-imbalance, thus detecting shilling in the system. Lifang Ren et al. in [19] combined SVM with collaborative filtering to improve prediction precision as far as possible, which meant that SVM was used to filter out services users might dislike, and services listed on top N would be recommended according to preference . Problems related to recommendation precision and efficiency are generally caused by data missing, and scholars have proposed different solutions to these problems. In [16] Mehrbakhsh Nilashi et al. made the SVM and multi-criteria collaborative filtering (MC-CF) as a combination to improve the recommendation precision. Yeounoh Chung et al. in [4] used the SVM to find personalized experts, then these experts will be recommended to different users. In [12] Zhan Li et al. used multiple-kernel SVM to recommend new videos, which can relieve the problems of data sparsity and item cold start . Therefore, SVM-based collaborative filtering proposed in the paper is of significance in promoting recommendation precision and efficiency.

3 SVM-based collaborative filtering

3.1 Commodity information acquisition

In order to verify the actual effect of improved collaborative filtering on E-commerce industry, Python-based Scrapy is adopted in the paper to acquire online commodity information on Taobao, mainly including commodity name, commodity information and user's comments on it. Commodities on Taobao are graded by a 5-star marking system, and in the paper, it is shifted into a 5-point marking system. In addition, according to a huge amount of comments, Taobao translates them into characteristic value of commodities by semantic analysis, such as good stuff, fast logistics, etc. In the experiment, 3,4000 pieces of data are acquired and part of them are demonstrated in Table 1.

3.2 SVM-based classification

The data set in the experiment mainly includes marks and characteristic values of commodities. A 2500-dimension vector set is built based on the data, and the characteristic value which is nonexistent will be filled with 0. Suppose that data set of commodities on Taobao is $\{(x_i, y_i)\,|i = 1, 2, ...n\}$; the data set of commodities available for recommending is$\{x_i|i = 1, 2, ..., n\}$; $x_j = (x_{j1}, x_{j2}, ..., x_{jk})$ and $x_i = (x_{i1}, x_{i2}, ..., x_{ik})$ are characteristic attributes of set i and set j; $y_i \epsilon \{-1, 1\}$ is the output type. $y_j = -1$ means the commodity has negative feedback; $y_j = 1$ means the commodity has positive feedback. SVM builds classification model with commodity data set $\{(x_j, y_j)\,|j = 1, 2, ...n\}$ to find optimal hyperplane $g(x) = \langle w \cdot x\rangle + b = 0$. The corresponding optimization of SVM in the experiment is:

$$\underset{\alpha}{max} L(w, b, \alpha) = \sum_{j=1}^{n} \alpha_j - \frac{1}{2}\sum_{j=1}^{n} \alpha_i \alpha_q y_j y_q K(x_j x_q)\, s.t. \sum_{j=1}^{n} y_j \alpha_j = 0; 0 \le \alpha_j \le C \qquad (1)$$

$k(\cdot)$ is the radial basis function; optimal solution α_j^* can be obtained by optimization in formula (3-1) and thus figuring out the solution to the original question. $w^* = \sum_{j=1}^{n} \alpha_j^* y_j x_j$. Therefore, the classification decision function of optimal hyperplane definition can be expressed

Table 1: Data of commodity information

No.	Name	Price	Mark	Characteristic Values
1	Vero Moda 2018 autumn new chic knitwear	599	4.9	comfortable (179) pretty (134) I like it (85) I haven't wear it (59) good color (42) standard size (39)
2	Lin Shi Mu Ye cloth sofa bed	5960	4.9	skilled installation (1394) not bad quality (888) fast logistics (484) cost-effective (363) high price/performance ratio (104) no color difference (42) bad quality (6)
3	loose jeans for man in fall and winter	468	4.8	great quality (6681) comfortable (4132) suitable size (2816) good service (2585) good to wear (2366) cheap and fine (1996) thin cloth (613)
4	NTMPBINS suitcase with universal wheel	1658	4.8	good quality (721) good service (273) good style (270) fast logistics (222) no color difference (195) smooth wheels (122) fair quality (51)
5	excerpts from Historical Records with annotations	298	4.9	not bad (79) clear printing (61) good quality (54) cost-effective (50) thick paper (49) fast logistics (40)
6	2018 autumn new black dress with paillettes and tassels	513	4.7	good quality (193) beautiful (192) brighten your skin (192) soft cloth (153) non-deformation (135) new style (127) fair quality (39)

as below:

$$f\left(x\right) = sgn\left(g\left(x\right)\right) = sgn\left(\sum_{j=1}^{n} \alpha_j^* y_j k\left(x_j, x\right) + b^*\right) \tag{2}$$

According to formula (2), all commodities can be divided into two categories. When $f(x) = -1$, the commodity has negative feedback; when $f(x) = +1$, the commodity has positive feedback. Commodities fall into two categories through the classification of SVM. Then the data representing users' dislike are eliminated and only data representing users' affection are reserved.

3.3 Comprehensive grade calculation

After the classification with SVM, the paper in this part will only make sentimental analysis of commodities with positive feedback and obtain quantified sentimental intensity, and then solve sentimental intensity after sentimental analysis and commodity marks with the method of weighted average, finally obtaining comprehensive grades. Sentimental matching algorithm is adopted to match commodity comments with the ontology base and work out corresponding sentimental intensity. Steps for computing the sentimental intensity of commodity comments are as follows: Firstly, normalize the word frequency of characteristic values in commodity comments as showed in Table 2.

Table 2: Normalization of word frequency of commodity characteristic values

ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Description	good stuff	good service	fast logistics	good content	official edition	good packaging	clear printing	high-quality paper	fair quality
Word Frequency	480	207	129	117	86	84	72	70	34
Normalization	0.3752	0.1618	0.1008	0.0914	0.0672	0.0656	0.0562	0.0547	0.0265

Next, use sentimental words matching algorithm to match commodity characteristic values with ontology base to obtain sentimental intensity and polarity of the corresponding characteristic value, and the formula to calculate the comprehensive sentimental intensity of a commodity's comments is:

$$intensity\left(f_i\right) = K_1 P\left(w_{i1}\right) T\left(w_{i1}\right) + ... + K_n P\left(w_{in}\right) T\left(w_{in}\right) \tag{3}$$

w_{ij} represents sentimental words in commodity comments; $P\left(w_{i1}\right)$ and $T\left(w_{i1}\right)$ represent sentimental intensity and tendency of sentimental words respectively. When the polarity of $w_{ij} = 1$, $T\left(w_{i1}\right) = 1$. When the polarity of $w_{ij} = 0$,$T\left(w_{i1}\right) = 0$. When the polarity of $w_{ij} = 2$, $T\left(w_{i1}\right) = -1$. k_n is the corresponding normalized value of the characteristic value n . Finally, calculate the comprehensive grade of the commodity with formula (4) $\alpha = 5$; g_i is the mark of the commodity:

$$CE = \alpha f_i + (1 - \alpha) g_i \tag{4}$$

3.4 Similarity calculation and recommendation

The data processed in Table 3 are saved in matrix $R(U, I)$. U represents the number of users in the recommender system; I represents the number of items in the recommender system; r_{ij} is the mark of item I_j given by user u_i , representing users' affection for the item.w_{ij} is the similarity between item I_i and item I_j . Cosine similarity is adopted in the paper to measure the similarity between users or items. It measures similarity by the included angle between vector quantities.

As it does not take users' different rating scales into consideration, the cosine similarity in the modeling is improved by deducting users' average mark $\overline{r_u}$. Similarity w_{ij} between item I_i

Table 3: Example of comprehensive grade calculation-programming Python-from introduction to practice

Comment	**good stuff**	**good service**	**fast logistics**	**good content**	**official edition**	**good packaging**	**clear printing**	**high-quality paper**	**fair quality**	
Sentimental Classification	PH	PH	PA	PH	PH	PH	PH	PH	NN	
Intensity	3	7	7	5	4	6	5	3	9	
Polarity	1	1	1	1	1	1	1	1	2	
Assistant Sentimental Classification	0	0	PH	0	0	0	0	0	0	
Intensity	0	0	3	0	0	0	0	0	0	
Polarity	0	0	1	0	0	0	0	0	0	
Frequency	0.3752	0.1618	0.1008	0.0914	0.0672	0.0656	0.0562	0.0574	0.0265	
Sentimental Mark	1.1256	0.1326	0.46368	0.457	0.2688	0.3936	0.281	0.1641	-0.477	
comprehensive sentimental intensity										3.80938

and item I_j can be expressed as

$$w_{ij} = \frac{\sum_{u\epsilon U(i,j)}(r_{ui} - \overline{r_u})(r_{uj} - \overline{r_u})}{\sqrt{\sum_{u\epsilon U(i)}(r_{ui} - \overline{r_u})^2 \sqrt{\sum_{u\epsilon U(j)}(r_{uj} - \overline{r_u})^2}}}. \tag{5}$$

Comprehensive grades of some commodities and users are firstly calculated with the formula in (3), as showed in Table 4. Then the similarity between commodities is calculated according to formula (5), and part of the results are demonstrated in Table 5.

Table 4: Comprehensive grades of some commodities

commodity|user	**user1**	**user2**	**user3**	**user4**	**user5**
Python** 1	4.57	2.18	3.18	0.60	3.56
suit*** 2	3.35	1.88	0.00	4.58	4.92
children's shoes**** 3	1.88	0.00	2.10	4.24	0.67
Anchor**** 4	0.00	1.50	3.94	0.00	1.38
nuts*** 5	1.16	2.96	1.58	4.14	4.85

SVM-based collaborative filtering obtains k nearest neighbors of the item to be recommended by calculating the similarity between items, and then use item similarity and users' records to figure out the popularity grade of the item to be recommended by weighted calculation, finally forming a recommendation list according to the rank of grades. User u's grade on item I_j can be expressed as:

$$P_{uj} = \sum_{I_i \epsilon I_u \bigcap S(I_j,k)} w_{ji} r_{ui} \tag{6}$$

Table 5: Similarity between commodities

	Similarity
Python**,suit***	-0.15
Python**,children's shoes****	0.29
Python**,Anchor****	-0.50
Python**,nuts***	0.15
suit***,children's shoes****	0.21
suit***,Anchor****	-0.80
suit***,nuts***	0.70
children's shoes****,Anchor****	-0.28
children's shoes****,nuts***	0.02
Anchor****,nuts***	-0.26

L_u represents the set of items user u like in the records; $S\left(I_j,k\right)$ represents the "k" item sets that are most similar to I_j . SVM is used to build positive-feedback set and negative-feedback set $L_i=\{L_i|r_{ij}=1\}$ is the positive-feedback set; $DL_i=\{L_i|r_{ij}=-1\}$ is the negative-feedback set; $B_i=\{L_i|r_{ij}=0\}$ is the unselected set.

In SVM-based collaborative filtering, the training set $X=\{\left(x_i,y_i\right)|x_i\epsilon L_i\bigcup DL_i,y_i\epsilon\left\{-1,1\right\}\}$, test set $TX=\{tx_i\epsilon B_i\}$.$y_i$is the classification tag of x_i; when $x_i\epsilon L_i$;$y_i=1$ when $x_i\epsilon DL_i$, $y_i=-1$. The main steps of SVM-based collaborative filtering are as follows: Initialization: training set $X=\Phi$, test set $TX=\Phi$,

- use Pytho-based Scrapy to acquire commodity information on Taobao.
- use SVM to divide commodities into positive-feedback commodities and negative-feedback commodities, and build training set X and test set TX .
- use training set X to train SVM classifier.
- use classifier to classify test set, filtering out negative items and reserving positive items.
- make weighted calculation of marks and comments of positive-feedback commodities to get comprehensive grades.
- use $f\left(x\right)$ to predict the popularity grades of the items, and rank them according to the predicted grades, forming the final recommendation list.

4 Experiment results and analysis

4.1 Data preparation

The experiment adopts Python-based Scrapy to acquire data of online commodity information on Taobao, which include 7 categories of commodities (clothing, books, appliances, digital products, mobile phones, shoes and bags) and about 34,000 pieces of detailed comments. There are 4000 pieces of data for each category, among which 2500 pieces are taken as the training set and the rest are used for testing. Hardware : Thinkpad E445, 3.3GHZ, 4GB RAM.

4.2 Assessment indicator

Predictive accuracy P represents the probability that the user may like an item in the recommendation list, which can show the accuracy of the recommender system. The formula to calculate predictive accuracy of recommender system is as follow:

$$P = \frac{1}{m}\sum_{u=1}^{m} P_u = \frac{1}{m}\sum_{u=1}^{m} \frac{|RL_u \bigcap TL_u|}{N} \tag{7}$$

Recall rate R represents the proportion of items users like in the recommendation list, which can show users' satisfaction degree with the recommendation results. The higher the recall rate is, the higher satisfaction degree users have.

The formula to calculate the recall rate of recommender system is as follow:

$$R = \frac{1}{m}\sum_{u=1}^{m} R_u = \frac{1}{m}\sum_{u=1}^{m} \frac{|RL_u \bigcap TL_u|}{TL_u} \tag{8}$$

F–Measure is used to assess the overall recommending performance of the algorithm.

A larger F-Measure means the stronger recommending ability of the algorithm. The formula to calculate F-Measure of the recommender system is as follow:

$$F = \frac{1}{m}\sum_{u=1}^{m} F_u = \frac{1}{m}\sum_{u=1}^{m} \frac{2 * P_u * R_u}{P_u + R_u} \tag{9}$$

RL_u is user u's recommendation item set; TL_u is the set of items user u like in the test set; "U" is the recommended item number; $RL_u = N$. P, R and F are three main indicators to assess the algorithm. A larger F factor means the stronger recommending ability of the algorithm.

4.3 Results and analysis

(1) *Comparative analysis with traditional recommendation algorithm*

For the fairness and objectivity of the experiment, data recommendation in the paper is carried out according to traditional item-based recommendation algorithm and SVM-based collaborative filtering respectively, so as to compare the effect of recommendation. Traditional item-based recommendation algorithm is on the basis of neighborhood k . In order to be fairer and more objective, the recommended performance of algorithm with different neighborhood k will be analyzed in this part to choose the optimal k value. Figure 1 is the recommended performance of item-based recommendation algorithm with different k values. Three lines in the figure represents predictive accuracy, recall rate and F-Measure with different k values. It can be seen from the figure that the optimal recommended performance exists when $k = 10$, so $k = 10$ is selected in the subsequent experiment.

Figure 2, 3 and 4 represent the variation tendency of predictive accuracy P , recall rate R and F-Measure of SVM-based collaborative filtering and traditional recommendation algorithm respectively when the recommended item number N is different. It can be seen from the figures that when $N < 20$, predictive accuracy P , recall rate R and F-Measure of SVM-based collaborative filtering are obviously better than those of traditional recommendation algorithm. Therefore, SVM-based collaborative filtering has a big advantage over traditional recommendation algorithm when $N < 20$, especially when $N = 5$.

In real situation of recommending, due to users' limited time and energy, too much recommendation is meaningless to users, and will affect their shopping experience on the contrary. Therefore, what should be considered is the situation when N is relatively small. When $N < 15$

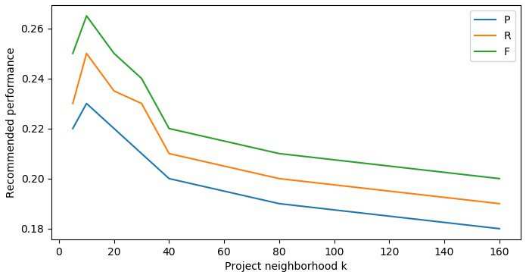


Figure 1: Recommended performance of item-based recommendation algorithm with different k values

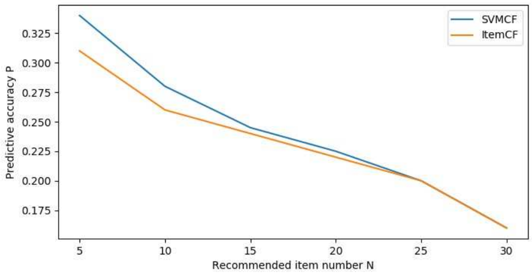


Figure 2: Predictive accuracy P of two models with different N values

, the recommended performance of SVM-based collaborative filtering is better than that of traditional recommendation algorithm, thus illustrating that when N is relatively small, SVM-based collaborative filtering can achieve better recommended performance.

To compare the efficiency of the two algorithms, during the performance comparison of traditional recommendation algorithm and SVM-based collaborative filtering in the paper, the experiment is divided into training stage and predication stage, and then time consumption of the algorithm in data set ML-100K is calculated respectively. Training time of traditional recommendation algorithm and SVM-based collaborative filtering is 33.89 seconds and 11.23 seconds respectively; test time of them is 250.01 seconds and 90.08 seconds respectively, thus demonstrating SVM-based collaborative filtering is more efficient.

(2) *Analysis of the recommended Performance of improved collaborative filtering*

Table 6 and Figure 5 are the recommended performance of SVM-based collaborative filtering

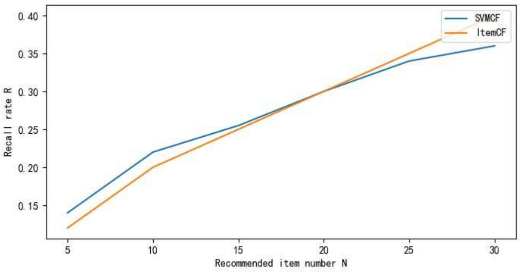


Figure 3: Recall rate R of two models with different N values

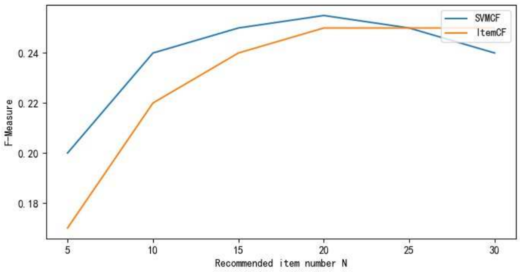


Figure 4: F-Measure of two models with different N values

with different N (number of different recommended items).

Table 6: Recommended performance of SVM-based collaborative filtering with different N

N (number of different recommended items)	**5**	**10**	**15**	**20**	**25**	**30**
F	0.2	0.24	0.25	0.255	0.25	0.24
R	0.14	0.22	0.255	0.3	0.34	0.36
P	0.34	0.28	0.245	0.225	0.2	0.16

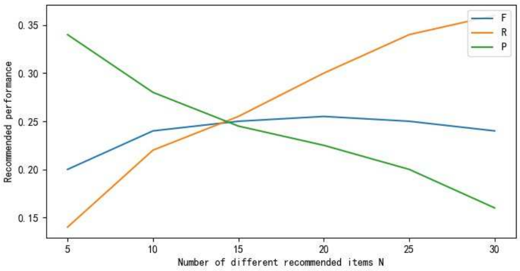


Figure 5: Recommended performance of SVM-based collaborative filtering with different N

It can be seen from the table and figure that with the increase of N, predictive accuracy P starts to decline and eventually levels off; recall rate R starts to increase dramatically; F-Measure shows the trend of increasing first and then decreasing, and finally levels off. These phenomena indicate that recommended performance does not always have a positive correlation with N . Therefore, an appropriate N should be selected to improve the recommended performance of SVM-based collaborative filtering. When $N = 15$, F and P have relatively high values, which demonstrates that the recommender system has the best effect at this point, and too many items for recommendation will be a burden for users on the contrary. Therefore, the algorithm in the paper has the optimal overall recommended performance when $N = 15$.

5 Conclusion

The efficiency of recommending commodities to users in the E-commerce industry is decided by the scale of recommended items and the performance of the recommendation algorithm. In the paper, SVM is adopted at first to classify items, and then items users may dislike are filtered out by negative-feedback information to reduce the scale of recommended items, which can not only significantly increase recommendation efficiency, but also decrease the disturbance of these items on recommendation and promote recommendation precision. After figuring out positive-feedback commodities, marks and comments are taken into consideration to obtain comprehensive grades by weighted average, thus making it more objective and indirectly improving precision. Finally, verification is conducted with the online data in the E-commerce industry. Therefore, the algorithm is of certain practical value for the E-commerce industry. Further studies can be carried out in the future with regard to how to further improve recommendation precision and efficiency of the recommendation algorithm in the situation of more recommended items.

Funding

This work was funded by Research supported by Beijing Logistics Informatics Research Base and Intelligent Emergency Logistics Linkage System of Public Emergencies in Beijing (Project No.18JDGLB019)

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

Bibliography

[1] Ahmad, A.S.; Hassan, M.Y.; Abdullah, M.P. et al. (2014). A review on applications of ANN and SVM for building electrical energy consumption forecasting, *Renewable and Sustainable Energy Reviews*, 33, 102-109, 2014.

[2] Barbieri, N. (2013). An Analysis of Probabilistic Methods for Top-N Recommendation in Collaborative Filtering, *Machine Learning & Knowledge Discovery in Databases-European Conference*, DBLP, 2013.

[3] Cheng, Q; Wang X; Yin, D. et al. (2015); The New Similarity Measure Based on User Preference Models for Collaborative Filtering, *IEEE International Conference on Information & Automation*, IEEE, 2015.

[4] Chung, Y; Jung, H.W.; Kim, J. et al. (2013). Personalized Expert-Based Recommender System: Training C-SVM for Personalized Expert Identification, *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, Springer, Berlin, Heidelberg, 2013.

[5] Du, Y,-P.; Yao, C.-Q.; Huo, S.-H. et al. (2017). A new item-based deep network structure using a restricted Boltzmann machine for collaborative filtering, *Frontiers of Information Technology & Electronic Engineering*, 18(05), 658-666, 2017.

[6] Goldberg, D.; Nichols, D.; Oki, B.M. et al. (1992). Using Collaborative Filtering to Weave an Information Tapestry, *Communications of the ACM*, 35(12),61-70,1992.

[7] Guo, G.; Zhang, J.; Thalmann, D. (1992). Merging trust in collaborative filtering to alleviate data sparsity and cold start, *Knowledge-Based Systems*, 35, 57-68, 2014.

[8] Hu, Y.; Peng, Q.; Hu, X. et al. (1992). Time Aware and Data Sparsity Tolerant Web Service Recommendation Based on Improved Collaborative Filtering, *IEEE Transactions on Services Computing*, 8(5), 782-794, 2015.

[9] Jindal, A.; Dua, A.; Kaur, K. et al. (2016). Decision Tree and SVM-Based Data Analytics for Theft Detection in Smart Grid, *IEEE Transactions on Industrial Informatics*, 12(3), 1005-1016, 2016.

[10] Li, G.; Ou, W. (2016). Pairwise probabilistic matrix factorization for implicitfeedback collaborative filtering, *Neurocomputing*, 204, 17-25, 2016.

[11] Li, H.; Hong, R.; Lian, D. et al. (2016). A Relaxed Ranking-Based Factor Model for Recommender System from Implicit Feedback, *IJCAI*, 1683-1689, 2016.

[12] Li, Z.; Peng, J.Y.; Geng, G.H. et al. (2015). Video recommendation based on multi-modal information and multiple kernel, *Multimedia Tools and Applications*, 74(13), 4599-4616, 2015.

[13] Liu, X. (2017). A collaborative filtering recommendation algorithm based on the influence sets of e-learning group's behavior, *Cluster Computing*, 1–11, 2017.

[14] Madadipouya, K. (2015). A Location-Based Movie Recommender System Using Collaborative Filtering, *Computer Science*, 5, 2015.

[15] Manek, A.S.; Shenoy, P.D.; Mohan, M.C. et al. (2017). Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier, *World Wide Web*, 20, 135-154, 2017.

[16] Nilashi, M.; Ibrahim, O.B.; Ithnin, N. et al. (2015); A multi-criteria recommendation system using dimensionality reduction and Neuro-Fuzzy techniques, *Soft Computing*, 19(11), 3173-3207, 2015.

[17] Nilashi, M.; Ibrahim, O.B.; Ithnin, N. (2014). Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and Neuro-Fuzzy system, *Knowledge-Based Systems*, 60(2), 82-101, 2014.

[18] Nasiri, M.; Minaei, B. (2016). Increasing prediction accuracy in collaborative filtering with initialized factor matrices, *Journal of Supercomputing*, 72(6), 2157-2169, 2016.

[19] Ren, L.; Wang, W. (2017). An SVM-based collaborative filtering approach for Top-N web services recommendation, *Future Generation Computer Systems*, S0167739X17300389, 2017.

[20] Sedhain, S.; Sanner, S.; Braziunas, D. et al. (2014). Social collaborative filtering for cold-start recommendations, 345-348,2014.

[21] Selakov, A.; Cvijetinovi, D.; Milovi, L. et al. (2014). Hybrid PSO–SVM method for short-term load forecasting during periods with significant temperature variations in city of Burbank, *Applied Soft Computing*, 16, 80-88, 2014.

[22] Su, H.; Lin, X.; Yan, B. et al. (2015). The Collaborative Filtering Algorithm with Time Weight Based on Map Reduce, *International Conference on Big Data Computing & Communications*, Springer, Cham, 2015.

[23] Uricar, M.; Timofte, R.; Rothe, R. et al. (2016); Structured Output SVM Prediction of Apparent Age, Gender and Smile From Deep Features, *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016.

[24] Wang, Z.; Liu, Y.; Chiu, S. (2016). An efficient parallel collaborative filtering algorithm on multi-GPU platform, *The Journal of Supercomputing*, 72(6), 2080-2094, 2016.

[25] Wei, J.; He, J.; Chen, K. et al. (2017); Collaborative filtering and deep learning based recommendation system for cold start items, *Expert Systems with Applications*, 69,29-39,2017.

[26] Yagci, A.M.; Aytekin, T.; Gurgen, F.S. (2017). Scalable and adaptive collaborative filtering by mining frequent item co-occurrences in a user feedback stream, *Engineering Applications of Artificial Intelligenceert Systems with Applications*, 58,2017.

[27] Zhang, F.; Gong, T.; Lee V.E. et al. (2016). Fast algorithms to evaluate collaborative filtering recommender systems, *Knowledge-Based Systems*, 96(C), 96-103, 2016.

[28] Zhang, D.W.; Xu, H.; Su, Z. et al. (2015). Chinese comments sentiment classification based on word2vec and SVM perf, *Expert Systems with Applications*, 42(4), 1857-1863, 2015.

[29] Zhao, P.X.; Gao, W.; Han, X. et al. (2019). Bi-objective collaborative scheduling optimization of airport ferry vehicle and tractor, *International Journal of Simulation Modelling*, 18(2), 355-365,2019.

[30] Zhao, P.X.; Luo, W.H.; Han, X. (2019). Time-dependent and bi-objective vehicle routing problem with time windows, *Advances in Production Engineering & Management*, 14(2), 201-212,2019.

[31] Zhou, W.; Wen, J.; Gao, M. et al. (2015). A Shilling Attack Detection Method Based on SVM and Target Item Analysis in Collaborative Filtering Recommender Systems, *International Conference on Knowledge Science*, 2015.

[32] [Online].http://www.askci.com/reports/20180201/0946472814827719.shtml.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 503-517, August 2019.

Energy Optimization for WSN in Ubiquitous Power Internet of Things

W. Hu, H.H. Li, W.H. Yao, Y.W. Hu

Wei Hu, Huanhao Li*, Wenhui Yao, Yawei Hu
School of Economics and Management,
Shanghai University of Electric Power, Shanghai 200090, China
*Corresponding author: 2625904776@qq.com

Abstract: This paper attempts to solve the problems of uneven energy consumption and premature death of nodes in the traditional routing algorithm of rechargeable wireless sensor network in the ubiquitous power Internet of things. Under the application environment of the UPIoT, a multipath routing algorithm and an opportunistic routing algorithm were put forward to optimize the network energy and ensure the success of information transmission. Inspired by the electromagnetic propagation theory, the author constructed a charging model for a single node in the wireless sensor network (WSN). On this basis, the network energy optimization problem was transformed into the network lifecycle problem, considering the energy consumption of wireless sensor nodes. Meanwhile, the traffic of each link was computed through linear programming to guide the distribution of data traffic in the network. Finally, an energy optimization algorithm was proposed based on opportunistic routing, in a more realistic low power mode. The experimental results show that the two proposed algorithms achieved better energy efficiency, network lifecycle and network reliability than the shortest path routing (SPR) and the expected duty-cycled wakeups minimal routing (EDC). The research findings provide a reference for the data transmission of UPIoT nodes.

Keywords: Ubiquitous power Internet of Things (UPIoT), energy-balanced routing, rechargeable wireless rechargeable network (WSN), routing algorithm, low power mode.

1 Introduction

During the "Two Sessions" (the annual sessions of China's top legislature and top political advisory body) in 2019, the State Grid Corporation of China (SGCC) promised to step up the construction of strong smart grid and ubiquitous power Internet of Things (UPIoT) and act as a hub, platform and sharer of energy information, turning itself into a world-class energy Internet enterprise with global competitiveness. Before long, the SGCC confirmed that its most urgent and critical task is to speed up the UPIoT construction, which requires overall planning and deployment. As its name suggests, the UPIoT refers to the application of the IoT technique in power systems. It is essentially a wireless sensor network (WSN) [15].

The WSN is a wireless network self-organized by numerous smart microsensor nodes, which collaboratively send the monitored data to the sink node for further processing. In the WSN, the safety, efficiency and speed of data transmission to the control center directly hinges on route optimization [9, 20]. Many sensors are deployed in the UPIoT. Under harsh natural conditions, the nodes in the rechargeable wireless sensor network (WSN) will enter the low power mode after running out of energy, if they can only capture a limited amount of energy. In this case, the operation time of network nodes depends on the energy-balancing effect of the routing strategy. Therefore, it is particularly important to manage the energy efficiency of sensors.

The common routing protocols fall into two categories: planar routing and clustering routing. The planar routing protocols include flooding protocol, SPIN (Sensor Protocols for Information via Negotiation) protocol, SAR (Synthetized Adaptive Routing) protocol, and directed diffusion mechanism [4]. In general, planar routing protocols are not applicable to the UPIoT, due to their fast energy consumption, slow response and high complexity. The typical clustering routing protocols are the LEACH (Low Energy Adaptative Clustering Hierarchy) algorithm, the TEEN (Threshold sensitive Energy Efficient sensor Network protocol) algorithm, and the PEGASIS (Power Efficient Gathering in Sensor Information Systems) algorithm [3, 16]. The latter two algorithms were extended from the LEACH. Compared with planar routing protocols, the clustering routing protocols feature low energy consumption, fast response and simple implementation. Unsurprisingly, this type of routing protocols has long been the research focus at home and abroad. Below is a brief review of the studies on clustering routing protocols.

In [12] Lee at al. proposes an energy-optimization clustering algorithm based on multiple factors, which introduces the fuzzy rule algorithm to cluster head selection, thus extending the network lifecycle. Reference [13] designs an energy-balanced non-uniform clustering routing algorithm, in which the cluster heads are selected by timer-based election. Reference [5] puts forward an energy-balanced routing algorithm based on message importance; the algorithm increases the success rate and reduces the delay of message delivery, as the route is determined by the forwarding income of messages. Reference [1] improves the low-energy adaptive clustering and stratification algorithm into an energy-balanced clustering routing protocol capable of adaptively adjust the cluster scale. Xia et al. in [19] proposed an energy equalization method based on hybrid transmission to avoid energy holes in wireless sensor networks. Hybrid transmission refers to the wireless transceivers of nodes in the network having different transmission powers. In a data collection sensor network, a wireless sensor near a sink node will deplete energy more quickly by receiving more data from a remote sensing node, and proposes to increase the transmission power of the distant node while probabilistically select the receiving node of the transmission to reduce the communication load of the node closer to the sink, delay the appearance of energy holes, and prolong the network life. The protocol ensure the balance of network energy by creating clusters of different scales based on the distance to sink node, residual energy and distribution density of network nodes, such that no low-energy node will be selected as cluster head.

To sum up, the existing energy-balanced clustering routing protocols face the following problems: the distributed algorithm in the protocols cannot converge to the global optimal solution; the energy consumption of information reception is ignored when analyzing the energy consumed in data transmission between network nodes; the routing algorithm searches for the optimal path, overlooking the possibility of multipath routing.

In light of the above, this paper proposes a multi-path routing algorithm based on link traffic distribution under the UPIoT. Specifically, the energy-balanced routing was achieved based on the predictable energy supplement, and the opportunistic routing was adopted to tackle the time-variation and loss of WSN channels. These measures, coupled with the natural load-balancing feature of the WSN, ensure the balance of the energy consumption among WSN nodes. The experimental results show that the proposed method can optimize the energy of rechargeable WSN, and prolong the lifecycle of the entire WSN.

2 UPIoT overview

The UPIoT is a smart service system that connects all things and all links in the power system, and supports effective human-machine interaction. This system can be implemented conveniently and flexibly to sense all kinds of states and process information in an efficient manner. The functions of the system are realized through modern information technologies like the mobile Internet and artificial intelligence, and advanced communication technologies. In fact, the UPIoT is an application of 5G (the 5th generation mobile networks) and the IoT in the power industry, linking up the human, machine and devices inside and outside the grid. As an extension of the energy Internet, the system can serve users, grids, power generation companies, governments and the entire society.

2.1 Construction principle

Unify standards and encourage innovation

Adhere to the unified data management, system construction must strictly follow the company's unified sg-cim data model and data collection, definition, coding, application and other standards, to ensure data sharing.

Adhere to the unified application interface, unified portal entrance, unified technology line, to ensure the application of horizontal interconnection, longitudinal through; We will combine top-level design with grassroots innovation, and encourage grassroots units to adapt measures to local conditions and take the lead.

Inheritance and development, precise investment

What is missing on the existing basis, integration and improvement, get through the data, avoid to scratch, find what you need and develop it, look for weak places and then strengthen it.

Technical and economic feasibility, vigorously promote; Pilot reserves that are technically feasible but economically to be assessed. If the investment is large and the effect cannot be seen in a short term, the demonstration will be limited in scope. Who to use, how to use, use frequency as the principle of whether to set up the project, to ensure accurate investment. Saving money is making money.

Intensive construction, sharing and co-construction

Coordinate the company's internal construction results, avoid repeated investment and development and pilot demonstration, promote the sharing and reuse of results, and give full play to the intensive effect.

All business terminals should fully consider the needs of all other majors, and be equipped with electrical side acquisition devices, communication resources, edge computing and data resources for cross-professional reuse, so as to promote the co–construction and sharing of all majors.

We will strengthen external cooperation in mature technologies, coordinate internal and external resources to advance them efficiently and ensure high-quality development.

Economic and practical, focusing on value

The key to the construction of ubiquitous power Internet of things is application. Full consideration should be given to practicality, economy and convenience of grassroots application. Only by working hard on practicality and practical effect can practical effect be achieved, so as to make front-line personnel better and more willing to use.

2.2 Specific content of construction

Build a comprehensive service platform for smart energy

With high-quality power grid services as the cornerstone and the entrance, we will make full use of the massive user resource advantages of state grid corporation to build a comprehensive intelligent energy service platform covering the government, terminal customers and the upstream and downstream of the industrial chain, and provide information docking, supply and demand matching, transaction matching and other services to attract users for emerging businesses. We will strengthen the building of common capacity centers for equipment monitoring, power grid interaction, account management and customer service, empower power grid enterprises and emerging business entities, and support the three-level smart energy service system for companies, regions and parks.

(1) Drainage: Integrate the external service application entrance and supply and demand information of various emerging businesses of state grid company, connect the energy efficiency service sharing platform of headquarter level enterprises, provincial customer side energy service platform, new energy big data platform, Internet of vehicles, photovoltaic cloud network, intelligent energy control system and other systems, and give full play to the large-scale agglomeration effect.

(2) Enabling: Integrating the common ability of state grid corporation for external services, providing all kinds of emerging business subjects with unified sharing services of network connection, monitoring, metering, billing, trading, operation and maintenance and other platform-based services.

Build an energy ecosystem

We should establish a standard system to support the interconnection of equipment, data and services, establish a normal cooperation mechanism with well-known enterprises, universities and scientific research institutions at home and abroad, integrate the upstream and downstream industrial chains, reconstruct the external ecology, promote the aggregate growth of industries, and create an energy Internet industrial ecosystem. We will build a national demonstration base for entrepreneurship and innovation, establish a mechanism for incubating emerging industries, serve small, medium and micro businesses, and actively foster new businesses, new formats and new models.

(1) Establishment mechanism: Make use of the innovation and entrepreneurship platform, give full play to the supporting service role of "collaborative sharing of needs, resource sharing, crowdfunding and crowdsourcing", pool innovation achievements, improve the mechanism of achievement transformation, and build the platform of achievement transformation.

(2) Promotion and transformation: Establish a special fund for achievement incubation, select outstanding achievements with broad market prospects and potential for state grid corporation operation, strengthen technical cooperation and capital cooperation, promote the industrialization of innovation achievements, and cultivate unicorn enterprises.

Cultivate and develop emerging businesses

Give full play to the SGC grid infrastructure, such as customers, data, brand unique advantages of resources, foster and develop a comprehensive energy service, Internet finance, big data operations, data inquiry, photovoltaic cloud network, three stand one, online financial supply chain and virtual power plants, based on a new type of energy services, intelligent manufacturing, chain block its core and the combination of 5G resources such as communication, tower emerging business, such as commercial operation implement emerging business "flowers", as a new major profit growth point of SGC:

(1) With the goal of serving the development of new energy industry, give full play to the unique resource advantages of state grid corporation, build new energy big data service platform, and carry out new business of new energy big data operation service.

(2) By collecting all kinds of data related to equipment operation, environmental resources, weather and climate, load energy consumption and so on at the power generation side, power

grid side and user side, we provide diversified services such as centralized equipment monitoring, equipment health management and energy efficiency diagnosis for power generation enterprises and comprehensive energy service providers.

Data sharing

Based on the unified data center and data model of the whole business, the data access transformation and integration are comprehensively carried out, the data standards are unified, the professional barriers are broken, and the data management system of the national network company is established. Create a data center, unified data call and service interface standards, and achieve data application service. Construction of an enterprise-level master data management system to support key tasks such as the transformation of multidimensional lean management systems.

Develop big data applications such as customer portraits, develop digital products, provide analytical services, and drive data operations:

(1) For the internal network of the company: To achieve equipment status warning, power sales and load forecasting, new energy generation power forecasting applications, etc. , to improve lean management.

(2) For the government industry: To achieve macroeconomic forecasting, energy conservation and emission reduction policy formulation, industry climate index analysis, big data credit and other services to support the government's efficient and accurate decision-making.

(3) For external enterprises: To achieve enterprise energy optimization recommendations, industry trend research, commercial location planning and other services to help enterprises save money and increase efficiency.

(4) For customers who use electricity: To achieve home energy optimization advice, quality service improvement and other services to enhance the sense of power users.

The UPIoT mainly consists of a sensing layer, a network layer, a platform layer, and an application layer. Based on widely distributed sensors, the sensing layer collects data in all stages of the grid, from generation, transmission, distribution to consumption, and conducts preliminary data fusion and calculation. The network layer fully utilizes modern communication technologies like 5G to set up a WSN that links up all data. The platform layer schedules the facilities and processes the big data in the network, making preparations for application. The application layer maintains the safe and stable operation of the grid and pursues an energy-efficient, smart and integrated power network.

3 UPIoT energy optimization

3.1 Rechargeable WSN

In the UPIoT, the data streams are transmitted via the rechargeable WSN [8]. The pattern of wireless charging determines the distance between the devices receiving and emitting the energy. The transmission distances, charging efficiency, power, frequency varies with the wireless charging patterns. This paper explores the charging mode based on radio waves. Capable of charging devices several tens of meters away, this wireless charging mode is suitable for WSNs with a large coverage. During wireless charging, the electromagnetic waves transmitted continuously from the power transmitter gradually attenuates with the increase of propagation distance and the change in the spatial environment. Thus, the author firstly examined the charging model of a single node in the WSN.

According to the electromagnetic propagation theory, the relationship between the electromagnetic wave power P_1 at any point in space and the transmitting power P_0 of the electromagnetic wave transmitter can be established as:

$$P_1 = \frac{GP_0}{\left(\frac{\lambda}{4\pi d}\right)^2} \tag{1}$$

where G is the combined gain of the receiving and transmitting antennas; d is the distance of the receiving point from the transmitting source; λ is the wavelength of the electromagnetic wave.

A rechargeable WSN that periodically collects data mainly consists of wireless sensor nodes, node links and power collection and transmission devices. Such a rechargeable WSN can be described as $G = (M, Z, P)$, where M is the set of all wireless sensor nodes, Z is the set of all node links and P is the set of power collection and transmission devices. M contains one sink node that collects data and uploads them to the processor and M-1 regular nodes that receive and transmit information. A node link exists between two wireless sensor nodes, if and only if the two nodes are within the transmission range of each other. In this case, the two nodes are called neighbors.

It is assumed that each wireless sensor node in the rechargeable WSN collects data at a constant rate, and the acquisition rate of node j is r$_j$. Let T be the lifecycle of the rechargeable WSN. For each cycle $t \in T$, the transmitting power of P is a constant denoted as $P_j^{fs}(t)$, with $j \in P$. For node j, the energy consumed to send a unit of data to node k can be denoted as e_{jk}^{fs}, and that consumed to receive a unit of data as e_k^{js}. The total amount of data transmitted by all wireless sensor nodes to the sink node via multiple paths can be denoted as $q_{jk}(t)$.

3.2 WSN opportunistic routing

Targeting unreliable networks, opportunistic routing increases network throughput through fewer transmissions and lower energy consumption, using multiple nodes in the next hop that may provide similar transmission quality [6,14]. The opportunistic routing is achieved in different ways, depending on the specific medium access control (MAC) protocol. The key to this routing method lies in the selection of candidate set. A candidate set is a set of candidate nodes, i. e. the next hop nodes that are qualified by the algorithm to forward packets. In a wireless network, the nodes in the candidate set for opportunistic routing are all selected from single-hop neighbors. The main reason is that, in a highly dynamic wireless network, it is only possible to maintain the accurate information of single-hop neighbors.

The topology of the candidate set for node S is shown in Figure 1, where the black box stands for the size of the area centered on S. It can be seen that this area contains the neighbors within two hops (h=2) from S: the first-hop neighbors include nodes A∼E and the second-hop neighbors include nodes f∼k. Among the second-hop neighbors, nodes g, h, i and k can interact with the nodes outside the area, and are thus called the exits of the area. Thus, the candidate set of node S can be described as$\{g, h, i, k\}$. Meanwhile, the first-hop neighbors and the second-hop neighbor that is not the exit of the area are service nodes for the candidate ones, and collectively form the set of service nodes $\{A, B, C, D, E, f, j\}$.

If node S needs to forward data, the shortest path tree will be constructed directly from the link state database for deterministic forwarding, if the destination falls in the said area; the forwarding nodes will be selected by the priority and dynamic forwarding probability of candidate nodes, if the destination falls outside the area.

Determine the priority of candidate nodes. For a candidate node, the lower the end-to-end cost of reaching the destination node when it is selected, the higher the corresponding priority, and the higher the corresponding forwarding probability. For example, for two candidate nodes, if one is in the direction of the destination node and the other is in the opposite direction of the destination node. Obviously, candidate nodes in the same direction have higher priority and

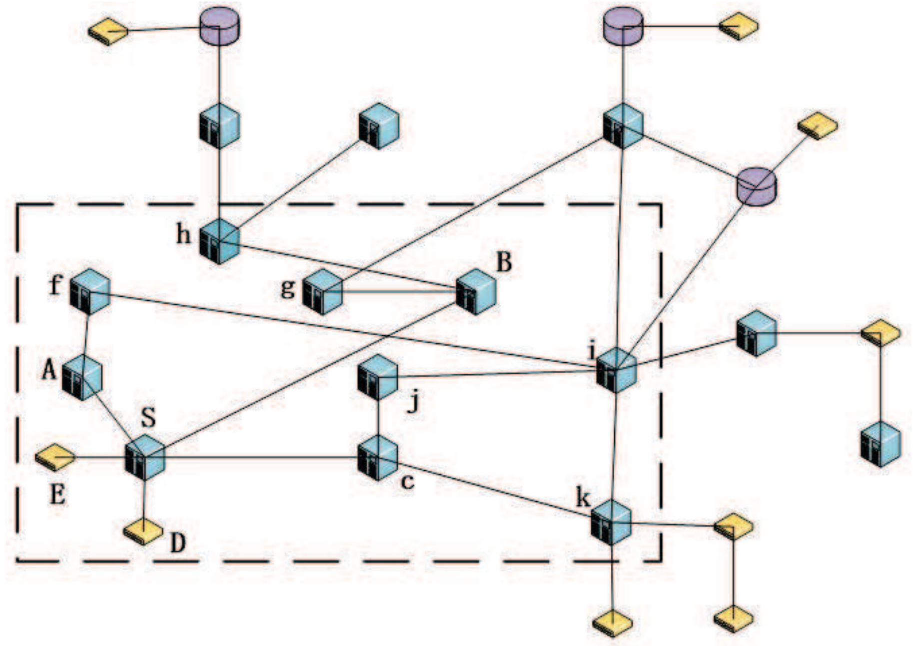


Figure 1: Topology of an example on candidate set

forwarding probability, and even worse, candidate nodes in the opposite direction can set the forwarding probability to zero. The cost corresponding to a candidate node includes two parts: the near cost and the far cost. The distant cost refers to the end-to-end cost expected by the candidate node to reach the destination node. According to link state database (LSDB), it is easy to calculate the approximate cost corresponding to the optimal path to a candidate node. The remote cost should be considered in two cases: if the destination node is located in the region with the candidate node as the center, the corresponding remote cost of the optimal path from the candidate node to the destination node can be obtained according to the LSDB of the candidate node; If the destination node is not in the region centered on the candidate node, considering the characteristics of opportunistic routing, this paper needs to consider the average end-to-end cost of all potential paths from the candidate node to the destination node, and take this as the remote cost of the candidate node.

Nodes have different forwarding probabilities under different conditions. For a particular destination node, not all candidate nodes (referring to candidate nodes after loop avoidance) have the opportunity to become the final forwarding node, because some candidate nodes will forward the packet to the more expensive path. In this paper, the factor α $(0 \leq \alpha \leq 1)$ is used to represent the proportion of candidate nodes with non-zero forwarding probabilities in the total candidate nodes: $\alpha = 1$ indicates that each candidate node has a non-zero forwarding probability and has the opportunity to become the final forwarding node. In this case, the performance of load balancing is the best; when $\alpha \leq 1/\mathrm{NS}$ (NS is the number of sending node S candidate nodes), the opportunity route is degraded into a deterministic route, and the candidate node with the least end-to-end cost is determined as the final forwarding node.

4 Energy optimization model and routing algorithm for the UPIoT

4.1 Energy balance analysis and routing algorithm

For a WSN, it is assumed that each node i is charged by the same device at a time. To ensure the energy sufficiency of node i in each cycle t, the node should select the device with the highest charging power in each cycle. Let the distance between node i and the power transmitter be $d_{i,r}$. Then, the charging power of node i in cycle t can be expressed as:

$$P_i^{\max}(t) = \max_{r \in R} \left\{ \frac{GP_r}{\left(\frac{\lambda}{4\pi d_{ir}}\right)^2} \right\} \tag{2}$$

According to the rechargeable WSN model, the data flow of a wireless sensor node i consists of three parts: the data stream flowing into node i $Q_i^{in}(t)$, the data stream flowing out of node i $Q_i^{out}(t)$, and the data stream generated by node i itself $e_i t$. In a traffic-stable UPIoT network, the final data will all enter the sink node. In this case, each node i in the network should satisfy:

$$Q_i^{in}(t) + e_i t = Q_i^{out}(t) \tag{3}$$

In other words, the traffic of network nodes is dynamically balanced. The data flow rate can be expressed as:

$$\sum_{i,r \in L_k} q_{i,r}(t) + r_i = \sum_{k \in L_k} q_{i,k}(t) \tag{4}$$

where L$_k$ is the set of neighbor nodes that forward data for a randomly selected node in the network.

Here, it is assumed that the wireless sensor nodes collect monitored data at a low frequency and low power consumption. Without considering the power consumed by the nodes in data acquisition, the energy consumption rate of node i in cycle t can be calculated by:

$$V_i = e_{jk}^{fs} \cdot \sum_{i,r \in L_k} q_{i,r}(t) + \sum_{k \in L_k} e_k^{js} q_{i,k}(t) \tag{5}$$

It is also assumed that, the sum of the energy received by node i from the power transmitter in cycle t ($E_i(t)$) and the remaining power of the battery in cycle $t-1$ equals the energy consumption of node i in cycle t, i. e. the total energy for data transmission and reception of node i. Then, the operation time of node i in cycle t can be expressed as:

$$T_i(t) = \frac{E_i(t) + E_i(t-1)}{V_i} = \frac{E_i(t) + P_i^{\max}(t-1) \cdot t_r}{e_{jk}^{fs} \cdot \sum\limits_{i,r \in L_k} q_{i,r}(t) + \sum\limits_{k \in L_k} e_k^{js} q_{i,k}(t)} \tag{6}$$

If $T_i(t) \geq t$, then the battery has surplus energy in cycle t and can support node i to operate beyond the cycle; otherwise, the energy received by node in cycle t, plus the remaining power of the battery, cannot support the normal operation of node i through cycle t. The relationship between $T_i(t)$ and t determines the amount of remaining power of the battery for node i in cycle t. An energy-balanced routing strategy must ensure that the remaining power of the battery for node i is neither zero or above the maximum battery capacity in any cycle. In actual application, it is possible in any cycle that the energy stored in a node is insufficient to support data transmission and reception at the node, calling for the design of an energy-balanced routing protocol. For any node with insufficient energy, the remaining energy in the previous cycle can be regarded as zero.

Definition 1 (Network lifecycle)**.** The network lifecycle in cycle t refers to the operation time of the rechargeable wireless sensor node which is shorter than that of any other node involving in data monitoring and transmission.

The maximization of network lifecycle, i. e. the balancing network energy through traffic distribution of each sensor node in each cycle t, can be described as:

$$\max_{q_{i,r}(t)}\{\min_{i\in M} T_i(t)\} \tag{7}$$

$$s.t.\ T_i(t) = \frac{E_i(t) + P_i^{\max}(t-1)\cdot t_r}{e_{jk}^{fs}\cdot \sum\limits_{i.r\in L_k} q_{i,r}(t) + \sum\limits_{k\in L_k} e_k^{js} q_{i,k}(t)}$$

$$\sum_{i,r\in L_k} q_{i,r}(t) + r_i = \sum_{k\in L_k} q_{i,k}(t)$$

$$q_{i,r}(t) \geq 0$$

$$i, r, k \in N$$

where $E_i(t)$ represents the energy received by node i from the power transmitting device at stage t; $P_i^{\max}$ represents the charging power of node i during time t; e_{jk}^{fs} is the transmission energy consumed by node j to transmit one unit of data packet to node k; e_k^{js} is the received energy consumed by j to receive one unit of data packets; $q_{i,r}(t)$ represents the data flow rate between nodes.

In essence, the network lifecycle maximization is a max-min linear programming problem for variable $q_{i,r}(t)$ under equal constraints.

This problem can be solved by the linear programming tool fminimax in the Matlab. In a rechargeable WSN, if a node consumes energy in data transmission and reception faster than it collects energy, then the node will die within a limited time; otherwise, the node energy will always surpass the battery capacity, causing a great waste of the energy provided by the charging device. The above calculation strikes a balance between the energy consumed in data transmission and reception and the energy collected from radio waves, thus maximizing the network lifecycle.

If applied to an actual network, the routing strategy with the global optimal energy balance, which is obtained through the optimization of global information of the network, has a certain impact on the energy consumption and lifecycle of the network [10, 11]. To further extend the network lifecycle, it is of practical significance to optimize the energy of the UPIoT based on the low power mode.

4.2 UPIoT energy optimization based on the low power mode

To further reduce the energy consumption and extend the lifecycle of the UPIoT sensor network, the low power MAC protocol is often adopted to turn off the wireless transceiver of inactive nodes, preventing unnecessary power consumption, that is, initiate the low power mode [15, 17].

Under this mode, the wireless sensor nodes can turn off the wireless communication module in the idle period, and enter the sleep mode. Then, the nodes can only send and receive data packet in the active state. The periodic sleep scheduling is an important aspect of the low power mode. By this strategy, the nodes fall asleep and wake up periodically at the preset time. The ratio of the sleep time to the scheduling cycle is called the duty ratio of the nodes in the low power mode. In the case of a low duty ratio, it is extremely rare for all nodes to wake up at the same time. To successfully transmit a data packet, the sender needs to send the introduction packet continuously until a receiver wakes up and confirms for its active state. If the sender only selects one next hop node, then it has to wait for the next scheduling cycle to send the remaining data packet after the end of the current active cycle of the receiving node.

In a sensor network with unreliable links, the sender must wait for several scheduling cycles, pushing up the data transmission time [2, 18].

The opportunistic routing allows a sender to select a candidate set of neighbors to forward data, rather than only a next hop node. In this way, it is possible to effectively shorten the waiting time, and reduce the transmissions to successfully send a data packet. In this section, the author attempted to optimize the network energy by setting up the candidate set of forwarding nodes for each network node, when the routing strategy cannot achieve the optimal energy balance [7, 10].

Suppose the sender needs to send a packet and consider multiple packets in the traffic distribution policy. The forwarding nodes in the candidate nodes of the sender s can be ranked as $f_s = \{a_1, a_2, \cdots, a_n\}$ by their wake-up sequence. Let $p_1, p_2, p_3 \cdots, p_n$be the quality of the sender s to each of the forwarding nodes. Then, the theoretical expected forwarding probability of node i in the candidate set can be determined by:

$$P(i) = \prod_{l=1}^{i-1} (1 - p_l) p_i \tag{8}$$

Equation (5) shows that the sender s fails to transmit data to any node in the candidate set until to node i. Next, equation (7) can be transformed to the selection of the optimal candidate set $f_i(t)$of forwarding nodes for node i within a rechargeable WSN in each cycle:

$$\max_{f_i(t)} \{\min_{i \in M} T_i(t)\} \tag{9}$$

$$s.t.\ T_i(t) = \frac{E_i(t) + P_i^{\max}(t-1) \cdot t_r}{e_{jk}^{fs} \cdot \sum\limits_{i.r \in L_k} q_{i,r}(t) + \sum\limits_{k \in L_k} e_k^{js} q_{i,k}(t)}$$

$$\sum_{i,r \in L_k} q_{i,r}(t) + r_i = \sum_{k \in L_k} q_{i,k}(t)$$

$$q_{ik}(t) = \frac{p_{ik}}{\sum\limits_{l=1}^{f_i} p_{il} \cdot \left(\sum\limits_{i,j \in f_j} q_{ij}(t) + r_i \right)}$$

$$i, r, k \in N$$

In equation (9), the traffic distribution depends on the following factors: the candidate set selected by opportunistic routing, the wake-up time, and the expected traffic considering node priority. The problem described by this equation can also be solved by the linear programming tool fminimax in the Matlab.

5 Case analysis

5.1 Experimental setup

The energy optimization routing protocol was simulated on the Matlab, aiming to verify its effects on common rechargeable WSN and low power rechargeable WSN. A total of $n - 1$ rechargeable WSN nodes were deployed randomly in an $800m * 800m$ field, with the sink node at the coordinates $(0, 0)$. The unreliable links were simulated by the free space propagation loss model. For simplicity, the charging scenario was simulated as a one-to-one mode. The charging device was placed at the height of $20m$ to supply energy to each node at a constant power. The energies consumed to transmit and receive a unit of data were respectively assumed as $e_k^{js} = e^R$ and $e^{fs} = e^T + \alpha d^\beta$. In addition, the scheduling cycle and duty ratio of the nodes in the low power mode were set as $2s$ and 60%, respectively. The specific simulation parameters are listed in Table 1 below.

Table 1: Simulation parameters

Sign	Definition	Value
n	Total number of network nodes	10, 30, 50 and 80
t_{sim}	Total simulation time	6h
α	Amplification factor	$80pJ \cdot b^{-1} \cdot m^{-3}$
β	Power index	4
Δt	Cycle	1h
t_r	Charging time	0. 5h
r	Sampling rate of WSN node	$80B \cdot \min^{-1}$
e^R	Energy consumption for receiving a unit of data of WSN node	0. 0567 mJ
e^T	Energy consumption for transmitting a unit of data of WSN node	0. 0233mJ

The proposed energy optimization (EOR) routing protocol is denoted as the EOR-L in the low power mode. In the simulation, the EOR was compared with the shortest path routing (SPR), and the EOR-L was contrasted with the expected duty-cycled wakeups minimal routing (EDC). In the SPR, the energy balanced routing is realized by taking the shortest path to the sink node based on the reciprocal of the ratio of the residual node energy to the rated battery capacity. In the EDC, the candidate set with the shortest expected delay is selected under the low power mode and opportunistic routing.

5.2 Evaluation standard

This paper uses the normalized network life cycle as the evaluation criterion to evaluate the strategy of routing energy balance with network as the target. The normalized network life cycle is defined as the ratio of the average working duration to the total period length of all nodes in the network (excluding the charging cycle). Taking Figure 2 as an example for specific explanation: node I in the first observation cycle [0, t] before a third charge of electricity from 0 to 1e, subsequent data transmission, forwarding operation, after t1 time consumption of energy, finished the work ahead of time; In the second observation period, I obtained 3e energy and consumed $3e - 0.7e$ energy. At the end of this period, there was still electricity left. Then the normalized network life cycle of I in the first two cycles is $(t1 - t/3 + 2t/3)/(4t/3)$.

5.3 Simulation results and analysis

The energy optimization quality of each routing protocol was evaluated by normalized network lifecycle (NNL), i. e. the ratio of the mean operation time of all network nodes to the cycle. Figure 3 shows the variation of the NNL with the number of nodes in the EOR and the SPR. It can be seen that the mean network lifecycle was shortened, as each node operated for

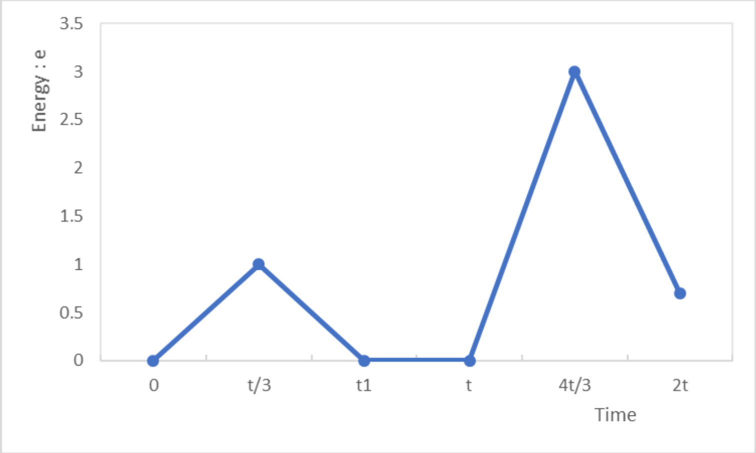


Figure 2: Battery energy cycle of node i

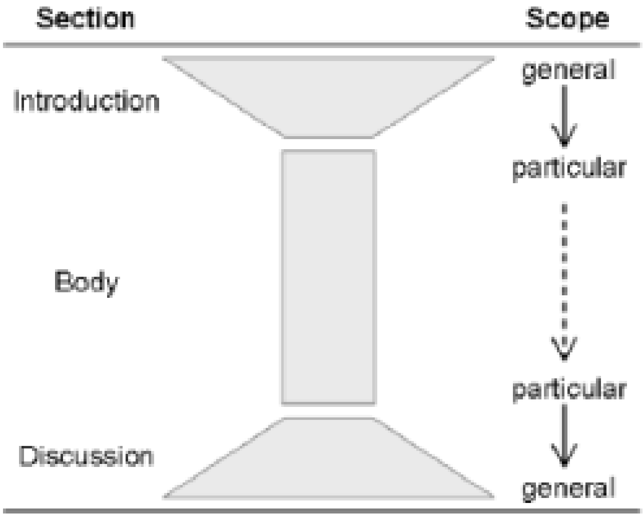


Figure 3: The variation of the NNL with the number of nodes

less time in each cycle, with the increase in the number of nodes. Both the EOR and the SPR saw a gradual decline in the NNL. Comparatively, the EOR achieved better energy optimization than the SPR. This is attributable to the fact that the EOR considers how the load distribution of multiple forwarding nodes, rather than a single next hop node, on the network optimization.

Figure 4 presents the variation of the NNL with the number of nodes in the low power mode. Similar to Figure 3, both the EOR-L and the EDC witnessed a declining trend in the NNL with the growth in the number of nodes. The EOR-L, which focuses on energy optimization, outperformed the EDC, which highlights time delay. Comparing Figures 3 and 4, it can be seen that the EOR-L had a better NNL than the EOR. Generally speaking, the EOR is a multi-path routing protocol, not an opportunistic protocol. By contrast, the EOR-L can enter the sleep mode when there is no task, thus saving the energy consumed in idle state.

Under the same scenario, the EOR led to a longer network lifecycle than the EOR-L. There are two possible reasons: our simulation only considers the energy consumed in data transmitting and reception; the EOR owns more information than the EOR-L, because it optimizes network energy using the global information of the network, while the latter adopts a distributed routing

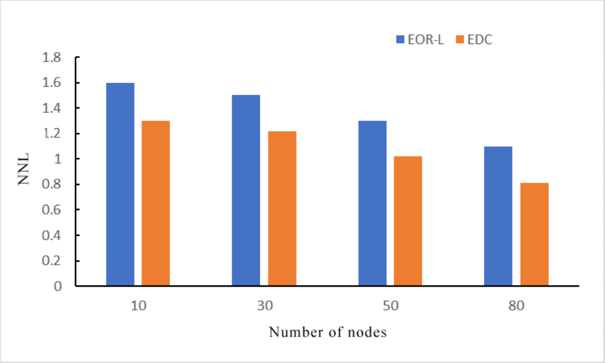


Figure 4: The variation of the NNL with the number of nodes in the low power mode

algorithm. Considering the complexity of the two algorithms, EOR uses a linear programming algorithm with a polynomial level of time complexity n; EOR-L adds constraints on the forwarding candidate set based on the former, the search space is larger than EOR, but still n polynomial level time complexity.

6 Conclusions

The UPIoT is the development trend of the energy industry and the priority of China's energy research. Under the UPIoT, the information transmission relies on the WSN containing multiple sensors. The WSN involves widely distributed power devices, which may need to work under harsh natural environment. For the energy-balanced routing of rechargeable WSN, this paper puts forward a multi-path routing protocol for the normal mode and an opportunistic routing protocol for the low power mode. Under the normal mode, the link traffic is planned through global linear programming for each forwarding node, according to charging rate, remaining battery power, and the energy consumed in data transmission and reception. Under the low power mode, the WSN lifecycle is greatly extended under the framework of opportunistic routing. Finally, Matlab-based simulation shows that the proposed routing protocols can effectively optimize network energy, reduce energy consumption of network nodes and enhance the network reliability. Next, we will investigate the routing problem of rechargeable wireless sensor networks in complex mobile scenarios.

Funding

This work is supported by the Humanities and Social Sciences Project of Chinese Ministry of Education (Grant No. : 17YJCZH062).

Bibliography

[1] Asha, G.; Santhosh, R. (2019). Soft computing and trust-based self-organized hierarchical energy balance routing protocol (TSHEB) in wireless sensor networks, *Soft Computing*, 23(8), 2537-2543, 2019.

[2] Awad, F.H. (2018). Optimization of relay node deployment for multisource multipath routing in Wireless Multimedia Sensor Networks using Gaussian distribution, *Computer Networks*, 145, 96-106, 2018.

[3] Caria, M.; Jukan, A.; Hoffmann, M. (2016). SDN partitioning: A centralized control plane for distributed routing protocols, *IEEE Transactions on Network and Service Management*, 13(3), 381-393, 2016.

[4] Chen, Y.; Xu, X.G.; Wang, Y. (2019). Wireless sensor network energy efficient coverage method based on intelligent optimization algorithm, *Discrete and continuous dynamical systems-series*, 12(4-5), 887-900, 2019.

[5] Chen, Z.G.; Yin, B.A.; Wu, J. (2018). Message of the importance of the opportunity to network based energy equilibrium routing algorithm, *Journal of communication*, 39(12), 91-101, 2018.

[6] Chowdhury, S.; Giri, C. (2019). Energy and Network Balanced Distributed Clustering in Wireless Sensor Network, *Wireless Personal Communications*, 105(3), 1083-1109, 2019.

[7] Ghazi, A.E.; Ahiod, B. (2018). Energy efficient teaching-learning-based optimization for the discrete routing problem in wireless sensor networks, *Applied Intelligence*, 48(9), 2755-2769, 2018.

[8] Gu, Y.; He, T. (2011); Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks, *IEEE Transactions on Mobile Computing*, 10(12), 1741-1754, 2011.

[9] Habib, M. (2019). Energy-Efficient algorithm for reliable routing of wireless sensor networks, *IEEE Transactions on Industrial Electronics*, 66(7), 5567-5575, 2019.

[10] Jayanthi, N.; Valluvan, K.R. (2018). Bio-inspired optimization routing technique using DNA sequencing algorithm for wireless sensor networks, *Wireless Personal Communications*, 101(4), 2365-2381, 2018.

[10] Khan, I.; Singh, D. (2018). Energy-balance node-selection algorithm for heterogeneous wireless sensor networks, *Electronics Journal*, 40(5), 604-612, 2018.

[11] Kulshrestha, J.; Mishra, M.K. (2018). Energy balanced data gathering approaches in wireless sensor networks using mixed-hop communication, *Computing*, 100(10), 1033-1058, 2018.

[12] Lee, J.; Kao, T. (2016). An improved three-layer low-energy adaptive clustering hierarchy for wireless sensor networks, *IEEE Internet of Things Journal*, 3(6), 951-958, 2016.

[13] Liu, C. (2015). Cluster head election strategy based on LEACH protocol in WSN routing algorithm and research, *Hangzhou university of electronic science and technology.*

[15] Liu, X.T.; Chen, Z.P.; Huang, Y.Y. (2019). A non-uniform clustering routing algorithm based on energy equilibrium, *Microelectronics and computer*, 36(2), 36-40, 2019.

[14] Liu, Y.; Wu, Y.; Chang, J. (2019). The diffusion clustering scheme and hybrid energy balanced routing protocol (DCRP) in multi-hop wireless sensor networks, *AD HOC and Sensor Wireless Networks*, 43(1-2), 33-56, 2019.

[15] Mittal, N. (2019). Moth flame optimization based energy efficient stable clustered routing approach for wireless sensor networks, *Wireless Personal Communications*, 104(2): 677-694, 2019.

[16] Shalabi, M.; Anbar, M.; Wan, T.; Khasawneh, A. (2018). Variants of the low-energy adaptive clustering hierarchy protocol: Survey, Issues and Challenges, *Electronics*, 7(8), 136, 2018.

[17] Sun, Z.; Wei, M.; Zhang, Z. (2019). Secure routing protocol based on multi-objective ant-colony-optimization for wireless sensor networks, *Applied Soft Computing*, 77, 366-375, 2019.

[18] Tabibi, S.; Ghaffari, A. (2019). Energy-efficient routing mechanism for mobile sink in wireless sensor networks using particle swarm optimization algorithm, *Wireless Personal Communications*, 104(1), 199-216, 2019.

[19] Xia, X.J.; Li, S.N.; Zhang, Y. (2015). Energy of mixed data transmission in one-dimensional sensor network Equilibrium, *Journal of Software*, 26(8), 1983-2006, 2015.

[20] Xiao, K.; Wang, R.; Deng, H. (2019). Energy-aware scheduling for information fusion in wireless sensor network surveillance, *Information Fusion*, 48, 95-106, 2019.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 518-529, August 2019.

Efficient Detection of Attacks in SIP Based VoIP Networks using Linear l_1-SVM Classifier

W. Nazih, Y. Hifny, W.S. Elkilani, T. Abdelkader, H.M. Faheem

Waleed Nazih*
1. College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, KSA
2. Faculty of Computers and Information Sciences, Ain Shams University, Egypt
*Corresponding author: w.nazeeh@psau.edu.sa

Yasser Hifny
Faculty of Computers and Information, Helwan University, Egypt

Wail S. Elkilani
1. Faculty of Computers and Information Sciences, Ain Shams University, Egypt
2. College of Applied Computer Science, King Saud University, KSA

Tamer Abdelkader
Faculty of Computers and Information Sciences, Ain Shams University, Egypt

Hossam M. Faheem
Faculty of Computers and Information Sciences, Ain Shams University, Egypt

Abstract: The Session Initiation Protocol (SIP) is one of the most common protocols that are used for signaling function in Voice over IP (VoIP) networks. The SIP protocol is very popular because of its flexibility, simplicity, and easy implementation, so it is a target of many attacks. In this paper, we propose a new system to detect the Denial of Service (DoS) attacks (i.e. malformed message and invite flooding) and Spam over Internet Telephony (SPIT) attack in the SIP based VoIP networks using a linear Support Vector Machine with l_1 regularization (i.e. l_1-SVM) classifier. In our approach, we project the SIP messages into a very high dimensional space using string based n-gram features. Hence, a linear classifier is trained on the top of these features. Our experimental results show that the proposed system detects malformed message, invite flooding, and SPIT attacks with a high accuracy. In addition, the proposed system outperformed other systems significantly in the detection speed.

Keywords: Machine learning, Support Vector Machines (SVMs), Session Initiation Protocol (SIP), VoIP attacks.

1 Introduction

Voice over Internet Protocol (VoIP) is a technology that enables the user to make voice or telephone calls over the Internet Protocol (IP) networks. Since the internet has been, and continues to be a prominent form of communication, the VoIP services are going to be a promising communication medium because of its low cost and added features. VoIP systems have two main functions: signaling function and media transmission function.

The most popular protocols developed for the signaling function are the Session Initiation Protocol (SIP) and $H.323$ protocol. SIP [19] is an application layer protocol to create, modify, and terminate real-time sessions between participants over an IP based network. Although $H.323$ protocol is more powerful [16], SIP is more popular because it is flexible, simple, easy to implement, and is based on ASCII messages (not binary messages as in $H.323$).

SIP is vulnerable to many attacks [33]. DoS attacks (i.e. malformed message and invite flooding) can disrupt the VoIP service partially or totally. In addition, SPIT attacks are usually used for products advertisement, harassment of subscribers, or convincing VoIP users to dial specific numbers. Recent statistics showed that spam calls result in huge man labor losses, especially for small business enterprises in the U.S. [32].

In this paper, we developed a machine learning (ML) system that decreases VoIP-SIP attacks using a linear SVM classifier. Our contributions are: (*i*) Introducing a novel approach for VoIP-SIP attacks detection using a fast linear SVM classifier; (*ii*) Using l_1 regularizer in the objective function that leads to sparse solutions[1]; (*iii*) Comparing our results with the published state-of-the art systems.

The rest of the paper is organized as follows. The next section introduces the related work, with a focus on the ML approaches. In section 3, we describe the proposed approach to detect VoIP-SIP attacks. Data is explained in section 4. Experimental setup is illustrated in section 5. Finally, section 6 concludes the paper and discusses future work.

2 Related work

Decreasing VoIP attacks is a hot topic of research in the last few years. Hosseinpour et al. [9] used a Finite State Machine (FSM) to extract parameters of the SIP traffic in normal conditions. These parameters are used with fuzzy logic to detect DoS attacks. Tsiatsikas et al. [27] detected DoS attacks which exploit the SIP message body. They built a Session Description Protocol (SDP) parser using 100 rules, which achieved a high accuracy.

Machine learning (ML) is an artificial intelligence approach that creates a model to recognize some patterns based on training examples. The ML task usually consists of three phases: choosing a learning algorithm, training the algorithm using the training dataset, and evaluating the algorithm performance by running it on another dataset (test-dataset). The detection of VoIP-SIP attacks using ML methods is introduced in many research work. Nassar et al. [14] extracted a set of 38 features from a slice of SIP messages, a SVM classifier decides if this vector is anomaly or not and issues an event, the event correlator uses a set of rules and conditions to filter the classifier events and generates alarms when necessary.

Akbar et al. [2] introduced Packet-based SIP Intrusion Protector (PbSIP) to prevent SIP flooding attacks and SPIT. PbSIP contains a packet-based analyzer that uses a set of spatial and temporal features to reduce the required processing and memory, features computation module, and Naive Bayes and J48 classifiers. In addition, Asgharian et al. [3] introduced a set of 18 statistics features calculated from SIP headers. They used a SVM classifier to evaluate the proposed features. Pougajendy et al. [17] used a subset of [3] features plus 2 new features, they evaluated the proposed features using a SVM classifier.

Rieck et al. [18] converted the SIP message to a high-dimensional vector space using n-gram tokens. They measured the Euclidean distance between a new message and a built model to detect anomalous messages. Tang et al. [24] proposed a prevention and detection system of SIP flooding attacks. They integrated a three-dimensional sketch design with the Hellinger Distance (HD) technique.

In [23] Su et al. extracted 23 features to detect SPIT attacks using k-nearest neighbor classifier. They added weight to each feature using a genetic algorithm. Vennila et al. [29] introduced two phases model; a SVM classifier to classify the traffic into VoIP and non-VoIP, and an entropy model to classify the VoIP traffic into flooding and non-flooding. Later, in [30] they proposed another two phases model to detect SPIT callers, which used Markov Chain, and incremental SVM classifier.

[1] By sparse solutions we mean that most of the parameters of the model are zeros.

In [25] Tsiatsikas et al. proposed an offline system to detect Distributed DoS (DDoS) attacks, they calculated the occurrence of 6 mandatory headers of SIP message, and implemented headers anonymization using HMAC. They tried 5 classifiers to find the best false alarm rate. Later, in [26] they proposed a real-time detection system and tried a group of DDoS scenarios.

Akbar et al. [1] used kernel tree analysis instead of features extraction, and a SVM classifier to detect malformed DDos attacks. In [21] Semerci et al. detected DDoS attacks using a change-point method which detects the change of Mahalanobis distance between successive feature vectors. If the change exceeds a threshold, the system labeled this as an attack. Kurt et al. [12] extracted a set of features from SIP messages and server logs. A Hidden Markov Model was used to relate these features to hidden variables, and a Bayesian multiple change model used these variables as change point indicators to detect DDoS flooding attacks. Le et al. [13] built a large data-set using a developed interface over a mobile application, which enables the user to label the malicious calls. They started with 29 features and reduced them to 10 features. Many machine learning models were tried (i.e. SVM and neural networks).

We observed a few issues in the developed systems described above. The feature extraction methods based on hand-crafted features are not generic, and tuned for specific datasets and attacks [3, 14, 23]. Hence, there is a need to develop a generic feature extraction method that is suitable for many attacks. Besides, the classification approaches based on distance measures and static threshold are not immune to noisy datasets [9, 29]. Moreover, the dual SVM methods are known to be slow due to the kernel calculations [1, 17]. Furthermore, low detection accuracy in general [21] or in case of low-rate attack [24] were observed. Lastly, rules-based systems require deep knowledge of SIP and numerous manual work [27].

These drawbacks led to the need for a system to detect VoIP-SIP attacks with high detection accuracy and a little processing time. To achieve this, we used the n-gram technique to extract features from SIP messages, and linear l_1-SVM to classify these messages into normal or attack.

3 Proposed approach

Detection of SIP attacks is formulated based on a ML approach. It consists of two steps. The first step is to project the messages into a high-dimensional space since they are more likely to be linearly separable than low-dimensional space [5], as illustrated in Figure 1. A method based on extracting n-gram tokens from a SIP message is used to generate the high-dimensional space. The second step is to use a linear SVM classifier with l_1 regularization to detect the SIP attacks. This classification algorithm optimizes the primal soft-margin objective function, and it is much faster than optimizing the dual objective functions with kernels that were used in the previous research [3, 14, 17].

3.1 Features extraction

In order to classify the SIP messages into normal or attack, the SIP messages are converted into numerical feature vectors. The features can be based on heuristics and domain knowledge as in [3, 23]. The disadvantage of this approach is that the generated features do not capture the diversity of the SIP messages, and they are highly tuned for specific attacks. Alternatively, they can be generated using a generic mathematical method like n-gram tokens as in [18]. The n-gram methods are widely used in speech and language processing [11] and in the network intrusion detection [31].

The SIP message is converted to a feature vector by moving a window of length n over the message and extracting all sub-strings (n-gram tokens). The length of the feature vector equals the number of unique n-grams in the training set. For each n-gram, we compute the number

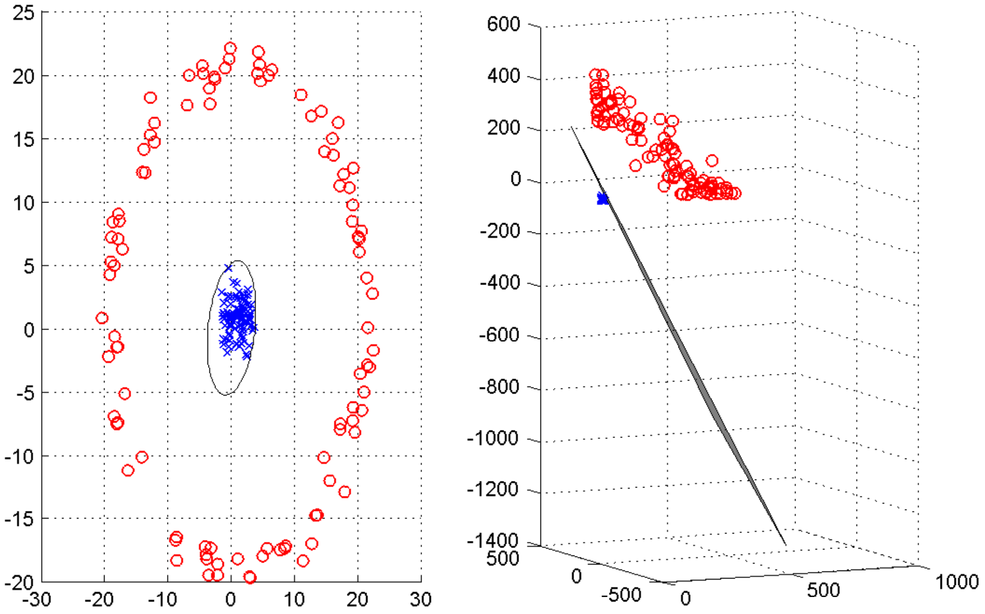
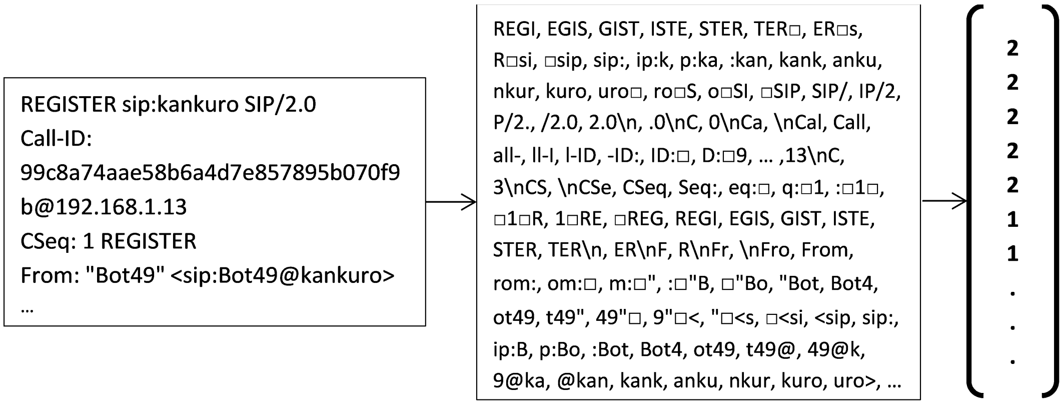


Figure 1: Moving to three-dimensional space, a nonlinear decision boundary for a two-dimensional classification problem becomes linear

of its occurrences in the message and use it to set its value in the feature vector. Figure 2 summarizes the features extraction process (n=4).



REGISTER sip:kankuro SIP/2.0
Call-ID:
99c8a74aae58b6a4d7e857895b070f9
b@192.168.1.13
CSeq: 1 REGISTER
From: "Bot49" <sip:Bot49@kankuro>
...

REGI, EGIS, GIST, ISTE, STER, TER□, ER□s, R□si, □sip, sip:, ip:k, p:ka, :kan, kank, anku, nkur, kuro, uro□, ro□S, o□SI, □SIP, SIP/, IP/2, P/2., /2.0, 2.0\n, .0\nC, 0\nCa, \nCal, Call, all-, ll-I, l-ID, -ID:, ID:□, D:□9, ... ,13\nC, 3\nCS, \nCSe, CSeq, Seq:, eq:□, q:□1, :□1□, □1□R, 1□RE, □REG, REGI, EGIS, GIST, ISTE, STER, TER\n, ER\nF, R\nFr, \nFro, From, rom:, om:□, m:□", :□"B, □"Bo, "Bot, Bot4, ot49, t49", 49"□, 9"□<, "□<s, □<si, <sip, sip:, ip:B, p:Bo, :Bot, Bot4, ot49, t49@, 49@k, 9@ka, @kan, kank, anku, nkur, kuro, uro>, ...

$$\begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 1 \\ 1 \\ . \\ . \\ . \end{bmatrix}$$

Figure 2: A part of a SIP message, its n-grams, and the occurrences of n-grams in the message

Although the extraction of features based on n-gram tokens provides a generic and an effective way for representing the SIP message, the length of the resulting feature vector is very large, which slows down the detection speed in the classification process. To overcome this problem, we set a cutoff hyper-parameter, and add to the feature vector the n-grams that exceed the cutoff.

3.2 Linear l_1-SVM classifier

Given a training set D that has m examples:

$$D = [(x_1, y_1), \ldots, (x_m, y_m)], \tag{1}$$

where y_i are either 1 or -1, each indicating the class to which the point x_i belongs (i.e. normal or attack). Each x_i is a d-dimensional real vector (i.e. the unique numbers of $n-$gram tokens in the training set). The soft-margin SVM classifier is computed by minimizing the *primal* objective function given by:

$$J = \min_w \left[\frac{1}{m} \sum_{i=1}^{m} \max\left(0, 1 - y_i(w \cdot x_i - b)\right) \right] + \lambda \|w\|^2, \tag{2}$$

where w is the vector of the parameters and $\max\left(0, 1 - y_i(w \cdot x_i - b)\right)$ is the *hinge* loss function. The term $\|w\|^2$ is the l_2 regularization penalty. The hyper-parameter λ is used to determine the trade-off between increasing the margin-size and ensuring that the x_i lie on the correct side of the margin.

The l_2 regularization penalty in Equation 2 does not lead to sparse solutions. The $l_1 = \|w\|$ regularizer or Lasso penalty is often used to increase the model sparseness since it can lead to solutions that have some elements with zero values [8]. In the proposed system, regularization is implemented by adding the l_1 norm penalty term to the hinge loss criterion (i.e. linear l_1-SVM classifier):

$$J = \min_w \left[\frac{1}{m} \sum_{i=1}^{m} \max\left(0, 1 - y_i(w \cdot x_i - b)\right) \right] + \lambda \|w\|, \tag{3}$$

In real implementation of Equation 3, the hinge loss is weighted by C:

$$J = \min_w \left[C \sum_{i=1}^{m} \max\left(0, 1 - y_i(w \cdot x_i - b)\right) \right] + \|w\|, \tag{4}$$

where the hyper-parameter $C = \frac{1}{m\lambda}$.

The solution of this objective function can be used to classify new points z:

$$c = \operatorname{sgn}(w \cdot z - b) \tag{5}$$

where c is the class identifier and b is a bias term.

The primal objective function in Equation 2 is commonly solved using a dual form with the Lagrangian [4, 28]. The dual form is given by:

$$J = \min_\alpha \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y_i \alpha_i k(x_i \cdot x_j) y_j \alpha_j, \tag{6}$$

$$\text{subject to} \sum_{i=1}^{m} \alpha_i y_i = 0, \text{ and } 0 \leq \alpha_i \leq \frac{1}{2m\lambda}; \text{for all i.} \tag{7}$$

where α are the parameters to optimize and $k(x_i \cdot x_j)$ is a kernel function. Some common kernels functions are: Polynomial (homogeneous): $k(x_i, x_j) = (x_i \cdot x_j)^d$, Gaussian radial basis function: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, for $\gamma > 0$, and Hyperbolic tangent: $k(x_i, x_j) = \tanh(\kappa x_i \cdot x_j + c)$, for some (not every) $\kappa > 0$ and $c < 0$. The new points z can be classified by computing:

$$c = \operatorname{sgn}\left(\left[\sum_{i=1}^{m} \alpha_i y_i k(x_i, z) \right] - b \right), \tag{8}$$

The main advantage of the dual form solution is producing a nonlinear classifier. However, it is slow in training (i.e. $O(m^2)$) and classification phases due to the usage of kernels [3, 14, 17]. On the other hand, the linear l_1-SVM classifier optimizes the primal soft-margin objective function, and is much faster than optimizing the dual objective functions with kernels.

The detection time is an important factor in the detection of attacks, and the dual form classifiers (i.e. Equation 8) may not be suitable for this task. The linear l_1-SVM classifier detection in Equation 5 is relatively fast. Hence, our main classifier is based on the primal form objective function solutions.

4 Data

To evaluate the proposed system, we need VoIP datasets. Unfortunately, real VoIP datasets are not available because of privacy concerns, so we used two generated datasets.

The first dataset was produced by INRIA [15]. They used two SIP proxy servers (i.e. Opensips and Asterisk), and VoIP attack tools to generate different scenarios of the invite flooding and SPIT attacks. The test-bed consists of a PC that acts as a server, two PCs generate the normal traffic using VoIP bots, and a PC that generates the attack messages. This dataset contains about 266,450 SIP messages.

In addition, the SIP-Msg-Gen tool [7] was used to generate the second dataset. It is a synthetic SIP message generator that generates normal SIP messages according to the SIP grammar defined in the RFC 3261 [19], and malformed SIP messages according to the SIP test messages defined in RFC 4475 [22]. The SIP-Msg-Gen tool can generate 14 different scenarios of the malformed SIP messages. All of these scenarios were used in the dataset generation. This dataset contains about 246,750 SIP messages.

For all experiments, we divided each dataset into three parts, 60% for training, 20% for cross validation, and 20% for testing. The training dataset was used to build the classification model. The cross validation dataset was used to tune the model hyper-parameters, and the test dataset was used to evaluate the final detection accuracy of the model.

5 Experiments

In this section, we evaluated the proposed approach using INRIA and SIP-Msg-Gen datasets. We projected the SIP messages into a high-dimensional space, and a linear l_1-SVM classifier was used for detection. In our proposed system we aim to achieve fast and high detection accuracy. Hence, we turned our attention to the primal form SVMs with l_1 regularization (i.e. linear l_1-SVM) to produce a sparse solution that will accelerate the detection process and decrease the number of active features. We compared the primal form SVMs with the dual form SVMs classifier. LibLinear [6] was used for the primal form l_1-SVM experiments. It is an open source library that solves large scale linear classification problems, and supports l_1 and l_2 regularizations. In addition, the LibSVM toolkit [10] was sued for dual form SVM experiments. All experiments are done in a machine with Intel Core i5 CPU, 3.2 GHz Quad-core and 8 GB RAM memory.

5.1 Evaluation

To evaluate the performance of our proposed model, we used F1 score [20]. It is the harmonic average of the precision and recall that takes into account the false positives and false negatives. The precision is the number of positive predictions divided by the total number of positive class predicted:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}, \tag{9}$$

and the recall is the number of positive predictions divided by number of positive class values.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \tag{10}$$

The F1 score is given by:

$$\text{F1} = 2 * \frac{\text{Precision * Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

where the precision and recall equally contributed into F1 score. The best F1 score is 1 and the worst is 0. The F1 score is usually more useful than accuracy, especially in case of different classes distribution.

The model sparseness was measured using the compression ratio (CR) criterion, given by

$$\text{CR}(C) = \frac{\#\text{Param - }\#\text{Pram(C)}}{\#\text{Param}} \tag{12}$$

where #Param is the number of features before the training process and #Pram(C) is the number of features after the training process, and is a function of the C parameter.

The target of the proposed system is to maximize the detection accuracy and minimize the required time for message classification. The average detection time $T_{\text{detection}}$ is computed as follows:

$$T_{\text{detection}} = \frac{\text{Features Extraction Time} + \text{Detection Time}}{\#\text{ Messages in Test Dataset}} \tag{13}$$

where the features extraction time is the total time required to compute the features for all messages in the test dataset and the detection time is the total time required to run the l_1-SVM classifier on the test dataset. The average train time T_{train} was computed using the same equation but over the *training* dataset.

5.2 Results

The proposed classifier was trained using the training dataset and different values of the hyper-parameter C were tried to achieve the best detection accuracy. The F1 score, $T_{\text{detection}}$, CR (C), and T_{train} were reported for these experiments.

Our first experiment was performed on INRIA dataset, the feature vectors were created using n-gram with n=4 and cutoff=5. All n-grams that existed in the training dataset more than 5 times are stored in the dictionary, the dictionary of INRIA dataset have 120,209 4-grams. This dictionary was used to create feature vectors for the cross validation and the test datasets. Then the primal form l_1-SVM classifier was tried with different C values.

In the second experiment, we used SIP-Msg-Gen dataset with the same hyper-parameters (n=4 and cutoff=5), and the dictionary has 290,166 4-grams. The size of this dictionary is bigger than the INRIA dictionary because the malformed messages usually contain random content.

The F1 Score for INRIA and SIP-Msg-Gen datasets with different C values is shown in figure 3. For INRIA dataset, the l1-primal SVM classifier achieved 100% detection accuracy at C= 0.0039063 using only 37 features out of the 120,209 in 0.737 milliseconds average detection time per message. For the SIP-Msg-Gen dataset, the 100% detection accuracy is achieved at C= 0.5 using 9,800 features out of the 290,166 features in 0.570 milliseconds.

Figure 4 shows the results of CR for INRIA and SIP-Msg-Gen datasets. Because a high detection accuracy was achieved with a few number of features, the compression ratio for both datasets is high. INRIA achived 99% compression ratio while SIP-Msg-Gen achieved 96% compression ratio.

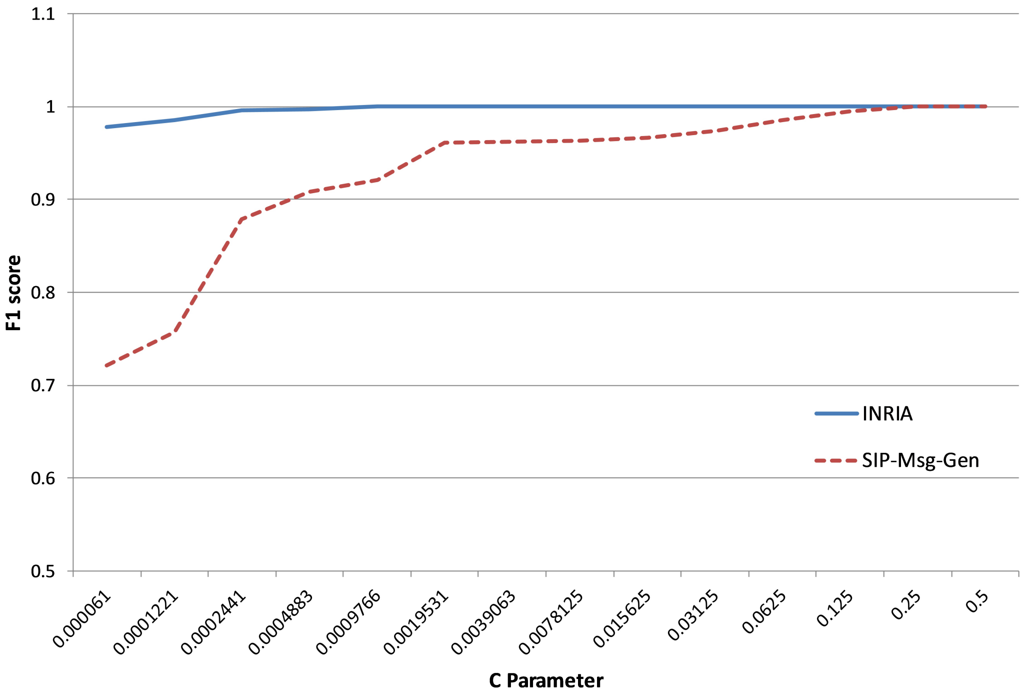


Figure 3: F1 score for INRIA and SIP-Msg-Gen datasets

The SIP-Msg-Gen dataset contains malformed messages, which usually have random content more than the invite flooding and SPIT messages in INRIA dataset. Hence, the l_1-SVM primal classifier needs more features with the SIP-Msg-Gen dataset to achieve high accuracy, and this leads to the low compression ratio with this dataset compared with the INRIA dataset.

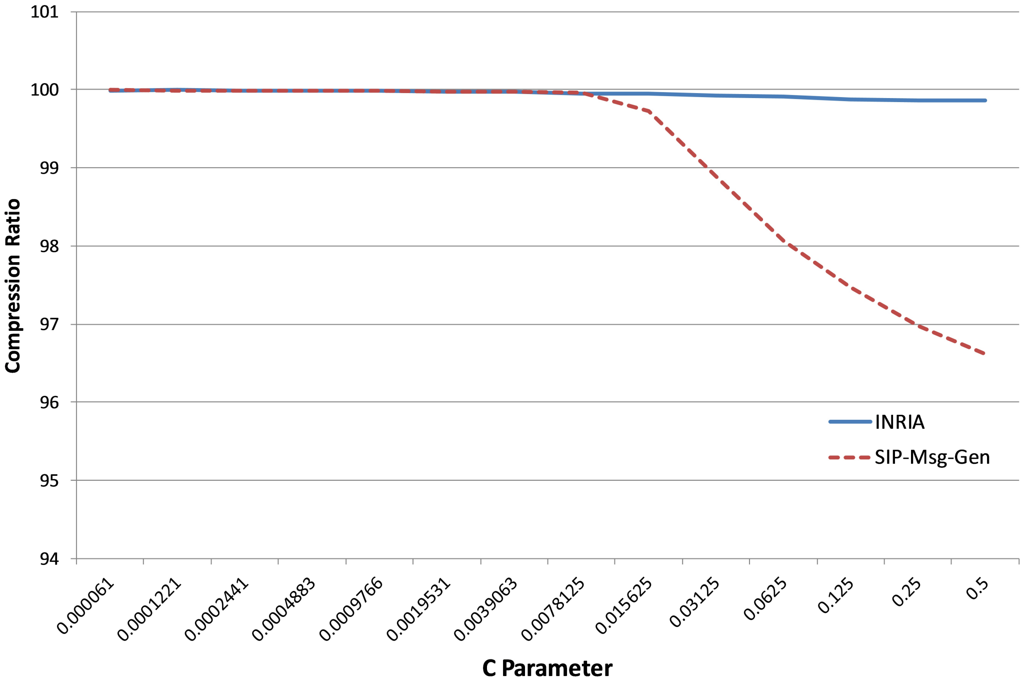


Figure 4: Compression ratio for INRIA and SIP-Msg-Gen datasets

Moreover, we calculated the proposed system throughput in megabits per second (Mbps) to measure the run-time performance. We achieved 2,700 Mbps using INRIA dataset, and 2,500 Mbps using SIP-Msg-Gen dataset which is considered a high throughput compared with previous work such as [18].

One of the important constraints for any learning algorithm is its scalability. When dealing with very large datasets, the dual form SVMs with kernels will be much slower than the primal form SVMs in detection and training. Although a fast detection is important, the fast training is also important especially in the case of the online adaption of the system.

Comparing between the dual form SVM with RBF kernel and the primal form SVM when both of them achieved 100% detection accuracy is given in Table 1. The primal form training time was about 17 times faster than the dual form using the INRIA dataset, and about 400 times faster with the SIP-Msg-Gen dataset. The detection time of the primal form was about 13 times faster than the dual form using the INRIA dataset, and about 100 times faster with the SIP-Msg-Gen dataset. This dual form setup is similar to the one used in [3].

Table 1: Comparison between the training time and the detection time for the primal and the dual form SVMs

Dataset	**SVM**	$T_{\mathbf{train}}$	$T_{\mathbf{detection}}$
INRIA	Primal	0.7445 ms	0.7370 ms
INRIA	Dual-RBF	13.0130 ms	10.1331 ms
SIP-Msg-Gen	Primal	0.5757 ms	0.5702 ms
SIP-Msg-Gen	Dual-RBF	219.5450 ms	57.0270 ms

Finally, we compared our linear l_1-SVM classifier to the state-of-the-art systems in Table 2. This comparison can only be considered as an indicator that the linear l_1-SVM classifier outperforms the other systems in terms of speed (detection time), because many factors such as the hardware configuration and the test dataset are different.

Table 2: Comparison between linear l_1-SVM and state-of-the-art systems.

Method	**Performance**	$T_{\mathbf{detection}}$	**Attacks**
linear l_1-SVM	F1 100% [1] F1 100%	0.7370 ms 0.5702 ms	Flooding and SPIT Malformed Msgs
Change Point [21]	F1 88%	0.76 ±0.45 sec	DDoS
Bayesian Change Point [12]	F1 95%	–	DDoS
Markov Chain and SVM [30]	Acc. 96.3%	15.145 ms	SPIT
Dual SVM [3]	Acc. 95-97%	–	Flooding
Sketch Design and Hellinger Distance [24]	Acc. 88% [2] Acc. 100%	–	Flooding
Dual SVM [17]	Acc. 99.9%	0.057-0.384 sec	Flooding
SDP Parser [27]	Acc. 100%	17-60 ms	Malformed Msgs
Dual SVM [1]	Acc. 99.9%	–	(D)DoS

6 Conclusions

In this paper, we proposed a machine learning system to detect the attacks of SIP based VoIP networks. The system projects the messages into a high-dimensional feature vector using n-gram

[1] F1 in the case of INRIA and SIP-Msg-Gen datasets respectively.

[2] Accuracy of low-rate and high-rate attack respectively.

tokens. In addition, a linear classifier to detect the SIP attacks (i.e. l_1-SVM classifier) is used for classification. Our work considers the fact that optimizing the primal soft-margin objective function is much faster than optimizing the dual objective function with kernels. Hence, we avoid the main drawback of the traditional dual form SVMs. Using the INRIA and SIP-Msg-Gen datasets, the proposed linear l_1-SVM classifier achieved competitive detection results to the other systems. Moreover, it is much faster than the state-of-the-art systems in the detection speed. Future work may focus on capturing a real VoIP dataset from a site under many VoIP attacks.

Bibliography

[1] Akbar, A.; Basha, S.M.; Sattar, S.A. et al. (2016). An intelligent SIP message parser for detecting and mitigating DDoS attacks, *Int. J. Innov. Eng. Technol*, 7(2), 1-7, 2016.

[2] Akbar, M. A.; Farooq, M. (2014). Securing SIP-based VoIP infrastructure against flooding attacks and Spam Over IP Telephony, *Knowledge and information systems*, 38(2), 491-510, 2014.

[3] Asgharian, H.; Akbari, A.; Raahemi, B. (2015). Feature engineering for detection of Denial of Service attacks in session initiation protocol, *Security and Communication Networks*, 8(8), 1587-1601, 2015.

[4] Cortes, C.; Vapnik, V. (1995). Support-vector networks, *Machine learning, Springer*, 20(3), 273-297, 1995.

[5] Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, *IEEE transactions on electronic computers*, 3, 326-334, 1965.

[6] Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J. et al. (2008). LIBLINEAR: A library for large linear classification, *Journal of machine learning research*, 1871-1874, 2008.

[7] Ferdous, R. (2012). SIP-Msg-Gen : SIP Message Generator, [Online]. Available: https://github.com/rferdous/SIP-Msg-Gen, Accessed on 8 May 2019.

[8] Friedman, J.; Hastie, T.;Tibshirani, R. (2001). The elements of statistical learning, *Springer series in statistics New York*, 1(10), 2001.

[9] Hosseinpour, M.; Hosseini Seno, S.A.; Yaghmaee Moghaddam, M.H. et al. (2016). An anomaly based VoIP DoS attack detection and prevention method using fuzzy logic, *Telecommunications (IST), 2016 8th International Symposium on. IEEE*, 713-718, 2010.

[10] Hsu, C.-W.; Chang, C.-C.; Lin, C.-J. et al. (2003). *A practical guide to support vector classification*, National Taiwan University, Taipei, 2003 (last updated 2016).

[11] Jurafsky, D.; Martin, J. H. (2014). *Speech and language processing*, Pearson London, 3-ed, 2019.

[12] Kurt, B. et al. (2018). A Bayesian change point model for detecting SIP-based DDoS attacks, *Digital Signal Processing*, Elsevier, 77, 48-62, 2018.

[13] Li, H.; Yildiz, C.; Ceritli, T.Y. et al. (2018). A Machine Learning Approach To Prevent Malicious Calls Over Telephony Networks, *arXiv preprint arXiv:1804.02566*, 2018.

[14] Nassar, M.; State, R.; Festor, O. (2008). Monitoring SIP traffic using support vector machines, *International Workshop on Recent Advances in Intrusion Detection*, Springer, 311-330, 2008.

[15] Nassar, M.; State, R.; Festor, O. (2010). Labeled VoIP data-set for intrusion detection evaluation, *Meeting of the European Network of Universities and Companies in Information and Communication Engineering*, 97-106, 2010.

[16] Packetizer, I. (2011). H. 323 versus SIP: A Comparison, [Online]. Available: http://www.packetizer.com/ipmc/h323_vs_sip, Accessed on December 2018.

[17] Pougajendy, J. and Parthiban, A. R. K. (2017). Detection of SIP-Based Denial of Service Attack Using Dual Cost Formulation of Support Vector Machine, *The Computer Journal*, Oxford University Press, 60(12), 1770-1784, 2017.

[18] Rieck, K.; Wahl S.; Laskov, P.; Domschitz, P. et al.(2008) A self-learning system for detection of anomalous SIP messages, *Principles, Systems and Applications of IP Telecommunications. Services and Security for Next Generation Networks*, Springer, 90-106, 2008.

[19] Rosenberg, J. (2002). SIP: Session Initiation Protocol, *IETF RFC 3261*, 2002.

[20] Sasaki, Y. (2007). The truth of the F-measure, *Teach Tutor mater*, 1-5, 2007.

[21] Semerci, M.; Cemgil, A. T.; Sankur, B. (2018). An intelligent cyber security system against DDoS attacks in SIP networks, *Computer Networks, Elsevier*, 136, 137-154, 2018.

[22] Sparks, R.; Hawrylyshen, A.; Johnston Avaya, A. et al. (2006). *Session initiation protocol (SIP) torture test messages*, 2006.

[23] Su, M.-Y.: Tsai, C.-H. (2015). Using data mining approaches to identify voice over IP spam, *International Journal of Communication Systems, Wiley Online Library*, 28(1), 187-200, 2015.

[24] Tang, J.; Cheng, Y.; Hao, Y. (2012). Detection and prevention of SIP flooding attacks in voice over IP networks, *INFOCOM, 2012 Proceedings IEEE*, 1161-1169, 2012.

[25] Tsiatsikas, Z.; Fakis, A.; Papamartzivanos, D. et al. (2015). Battling against DDoS in SIP: Is Machine Learning-based detection an effective weapon?, *12th International Joint Conference on e-Business and Telecommunications (ICETE)*, IEEE, 4, 301-308, 2015.

[26] Tsiatsikas, Z., Geneiatakis, D.; Kambourakis, G. et al. (2016). Realtime DDoS Detection in SIP Ecosystems: Machine Learning Tools of the Trade, *International Conference on Network and System Security*, Springer, 126-139, 2016.

[27] Tsiatsikas, Z.; Kambourakis, G.; Geneiatakis, D. et al. (2019). The Devil is in the Detail: SDP-Driven Malformed Message Attacks and Mitigation in SIP Ecosystems, *IEEE Access, IEEE*, 7, 2401-2417, 2019.

[28] Vapnik, V. (2013). The nature of statistical learning theory, *Springer science & business media*, 2013.

[29] Vennila, G.; Manikandan, M.; Aswathi, S. (2015). Detection of SIP signaling attacks using two-tier fine grained model for VoIP, *TENCON 2015-2015 IEEE Region 10 Conference*, IEEE, 1-7, 2015.

[30] Vennila, G.; Manikandan, M.; Suresh, M. (2017). Detection and prevention of spam over Internet telephony in Voice over Internet Protocol networks using Markov chain with incremental SVM, *International Journal of Communication Systems, Wiley Online Library*, 30(11), 2017.

[31] Wang, K.; Parekh, J.J.; Stolfo, S.J. (2006). Anagram: A content anomaly detector resistant to mimicry attack, *International Workshop on Recent Advances in Intrusion Detection*, Springer, 226-248, 2006.

[32] [Online]. Marchex. (2018). Spam Phone Calls Cost U.S. 2018 Small businesses half-billion dollars in lost productivity, Available: http://goo.gl/jTrgp3, Accessed on 10 March 2019.

[33] [Online]. Nettitude. (2015). VoIP Attacks on the Rise, Available: https://www.nettitude.com/uk/, Accessed on December 2018.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 530-539, August 2019.

Performance Evaluation and Comparison of Scheduling Algorithms on 5G Networks using Network Simulator

D. Perdana, A. N. Sanyoto, Y. G. Bisono

Doan Perdana*
School of Electrical Engineering
Telkom University, Bandung, Indonesia
*Corresponding author: doanperdana@telkomuniversity.ac.id

Aji Nur Sanyoto
School of Electrical Engineering
Telkom University, Bandung, Indonesia
ajinsanyoto@gmail.com

Yoseph Gustommy Bisono
School of Electrical Engineering
Telkom University, Bandung, Indonesia
bisono@telkomuniversity.ac.id

Abstract: In this research, we compared the Round Robin (RR) and the Proportional Fair (PF) algorithms for different user equipment density scenarios using voice and video traffic, to evaluate the key impact on performance of 5G mmwave network. This research simulated on NS3.27 with an integrated mmwave module. Based on the result, we found that the RR is a good choice for voice traffic. It has a throughput of 3.65% better than PF with similar fairness index. On the other hand, we found that the PF is the right choice for video traffic due to has better result for throughput. It has a throughput of 1.24% better than RR. For fairness index round robin has better result for voice and video traffic.

Keywords: Network simulation, 5G networks, scheduling algorithm, round robin, proportional fair.

1 Introduction

Increase of internet users give challenge for Information and Technology (IT) industry especially service providers in order to provide high quality and low latency service quality. According to the situation above, the telecommunications industry began to move to fifth generation technology (5G). Millimeter waves that have a frequency spectrum of 28 GHz - 30 GHz appear as a central technology in fifth generation technology (5G), because of their potential with wide bandwidth to achieve the large throughput required by future networks [20]. It has been proposed to be an important part of the 5G network to provide multi-gigabit communication services [17]. Research about mmwave generally uses 28-30 GHz, free-license bands at 60 GHz, and E-bands at 71-76 GHz, 81-86 GHz, and 92-95 GHz. [4]. Mid-infrared ELT Imaginer and Spectrograph (METIS) 2020 provides requirements for 5G technology to have end-to-end (E2E) latency below 10 ms, this cannot be achieved by previous technology [9]. A lot of developer have already innovate not only on the physical layer, but also on several layers. Data allocation for small packages on TDMA scheduling LTE systems is inefficient because the transmission process is sent at fixed 1 ms Transmission Time Interval (TTI). Flexible TTI which has a flexible TDMA structure has been proposed in the study [8]. The TTI variable system also has flexibility in scheduling resources, which can handle the characteristics of various networks efficiently. The concept of flexible TDMA is a solution, considering 5G has various types of services with very diverse traffic, ranging from applications, devices, and usage. According to all advantages, this flexible scheme must be handled by the right scheduling algorithm. In this research, we analyzed choice of the scheduler has a significant impact on performance of 5G mmwave network.

In this research, we analyzed choice of the scheduler has a significant impact on performance of 5G mmWave network. The module to simulate scheduler algorithm that has been adapted to the flexibe TTI concept has been done by [16] in Network simulator 3. All wireless network protocols have similarities in terms of message scheduling [11].

This paper compare two scheduler algorithms and analyze performance parameters such as delay, throughput, and fairness index. We simulate Round Robin (RR) and Proportional Fair (PF) to find the best performing schedulers that are applied to flexible TTI schemes. We analyze the effect of scheduler on network performance such as delay, throughput, and fairness index. Round Robin (RR) is a scheduler that provides resources for users without considering channel conditions. This is a simple procedure that provides fairness [14]. This algorithm works by rotating the queue process. Each process has the same time allotment that is equal to time quantum (q). If this quantum time runs out, the server will handle the next process. There is a matrics to set user priorities for resource blocks. Expressed with matrix m with user i in resource block k [5]. The value of the user metrics above compared with other user metrics during the system. Users with the largest matrices will be served first.

$$m_{i,k} = w_i(t - Ti) \tag{1}$$

Where notation on the (1) known as :
$w_i =$ priority value for every service for user i
$t =$ current time
$Ti =$ last time when user i was served

Proportional Fair algorithm has main purpose to balance between throughput and fairness among all the users [1]. Different from the previous algorithm, this algorithm considers the channel conditions in the calculation of the matrics.Then the proportional fair algorithm calculates based on the value of the average data rate and throughput in the previous metrics calculation. There is a matrix to set user priorities for resource blocks. Expressed with matrix m with user i in resource block k :

$$m_{i,k} = \frac{d_{i,k}(f)}{R_i(f)} \tag{2}$$

With $R_i(f)$ is average throughput of user i computed in subframe f, and $d_{i,k}(f)$ is Achieveable throughput user k in m resource block and f subframe which is a Shannon expression for the channel capacity as

$$d_{i,k}(f) = log \lfloor 1 + SNR_{i,k}(f) \rfloor \tag{3}$$

2 Related work

In [17] research has been carried out stating that, with the increase in cellular data demand, 5G exploited the Internet a large number of variations in the millimeter wave (mmWave) band to increase communication capacity. mmWave itself is suitable for 5G network devices, depending on the communication characteristics of mmWave capable of overcoming system complexity and design, interference management and spatial reuse, anti-blockage, and dynamics due to mobility. The fundamental difference between mmWave communication and other communication systems, namely in terms of opposing high propagation, directivity, and sensitivity to blockages. The characteristics of mmWave communication can be utilized as potential for mmWave communication, including integrated design and systems, interference management, spatial reuse, anti-clogging, and dynamic control. In [14] a TTI-based design analysis was conducted and focused on flexible TTI-based designs, in terms of how well they utilized the allocated radio resources, and found that flexible frame structures exceeded fixed structures in all traffic scenarios discussed, especially for small burst traffic. So it can be concluded that the flexible TTI scheme will be very suitable to be applied on mmWave communication.

In [16] a research has been conducted on the implementation and validation of the mmwave module in NS-3. They redesigned several layers because mmWave will require innovation not only in the physical layer, but also across all layers of the communication protocol stack to fully utilize high throughput, low latency capabilities and maximum performance. In [14] research has been carried out stating that, round robin performance and proportional fair scheduling provide good performance for downlink transmission mode. But for different transmission modes, proportional fair is able to provide good data rates. Although round robbin provides individual data speeds that are better compiled and far from eNodeB, the absolute value of this data speed is not as high as the proportional fair. Therefore proportional fair may still be a good choice.

In the [2] study conducted by B. Barakat et al. it was stated that the growth of wireless traffic and the very high demand for data levels from users encouraged researchers to improve the performance of Long Term Evolution-Advanced (LTE-A). To optimize it, package scheduling is done which is able to distribute radio resources among users to improve network performance and spectrum efficiency. In this case it has been proven that generalized Proportional Fair (GPF) Schedulers are better than conventional other schedulers. Several studies about scheduling algorithm on cellular technology has been done in LTE system. S. Ismail et al [12] have compared of several scheduling algorithms and evaluated in terms of throughput, delay, packet loss, and fairness index on vehicular environment for uplink transmission in LTE networks.

In reference [18] shows that, RR algorithm produces a good fairness index and has a poor throughput and has a delay. In contrast, the MT algorithm has good throughput and bad fairness index. The PF algorithm has increased fairness and throughput but has poor delays due to traffic requirements and channel condition independence. 3LHA has good throughput for P1 and P2 connections but makes P3 connections starving. In contrast, 3LHA requires fairness improvement.

In [13], Mohnish Jha et al. compared Round Robin, Priority Set Scheduler and Proportional Fair scheduler by transmitting real-time voice packages and best effort services with changes in the number of users using NS3.24. The simulation results show that the round robin scheduler is better QoS performance compared to the other two at uplink and downlink. Nevertheless, rarely research about scheduling algorithm on 5G network. Research about 5G mmwave can be done to evaluate cross-layer and end-to-end performance. Several studies about scheduling on 5G networks also have been researched before. K. Gomez on research [10], provided a comparative study of a different scheduling disciplines that can be used in future 5G especially on emergency communications for public safety. In addition to proposing a new disciplinary scheduler, simulation results.

3 Research method

The simulations on this research were performed on the Network Simulator 3.27 with an additional mmWave module. The mmave module is designed for end-to-end simulations of 3GPP style cellular networks.

Figure 1 shows the flowchart system. After designing the module in the NS3 environment,

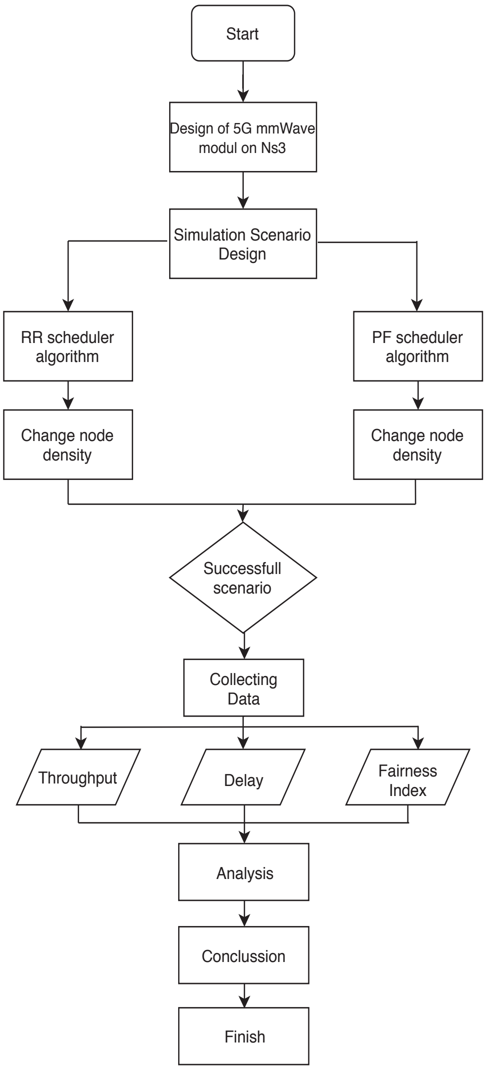


Figure 1: Flowchart system

the simulation design is adjusted to the scenario. The scheduling algorithm is implemented and simulated alternately. Changes on the number of nodes are set gradually from 20, 40, 60, 80, and 100 nodes. If the simulation is failed, the simulation scenario design will be reconfigured. If it is successful, we analyzed network performance such as throughput, delay, and fairness index. Throughput defined as the effective ability of a network in sending data. Throughput is the number of packets received in bits divided by the amount of delivery time [21].

$$Throughput = \frac{\sum Rx\ Packet\ Size}{Delivery\ Time} \tag{4}$$

Delay, defined as the time from packet send from sender to received in destination [19]. Average End to End Delay, which is the average time of delivering the data packet from the sender to the receiver [15]. The delay value starts calculated when the source starts sending packets and ends when the destination actually receives the packet. The delay can be constant, time-varying, or even random depending on the scheduler [3]

$$Delay = \frac{T_{rx} - T_{tx}}{\sum Rx} \tag{5}$$

Table 1: Simulation parameters

Parameter	**Quantity**
Mmwave Carrier frequency	28 GHz
Mmwave Bandwidth	1 GHz
Number of eNodeB	1
Number of User	20, 40, 60, 80, 100
User Mobilility	Constant Position
Datarate	Voice : 8 Kbps, Video : 386 Kbps
Packet Size	Voice : 20 bytes, Video : 240 bytes
Transport Layer	UDP
Scheduler Algorithm	Round Robin, Proportional Fair

Where notation on the (5) known as :
T_{rx} = Time of received packet on destination
T_{tx} = Time of packet send on source
$\sum Rx$ = Received packet

Fairness Index defined as the level of fairness of scheduling algorithms in schedule packages and allocation of resources to be sent. The theory and formula regarding the fairness index was revealed by [6]. Metrics of the formula are known as Jain's Fairness Index. Maximum value of this metrics is 1, where it indicates [7]

$$f(x) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2} \tag{6}$$

Where notation on the equation (6) known as : $f(x)$ = fairness index
n = number of user
x = Throughput user i

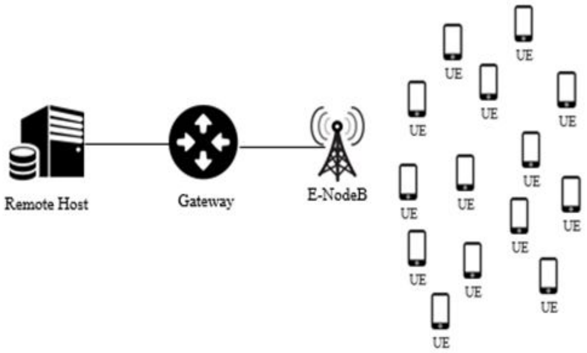


Figure 2: Simulation topology

The system design is shown in Figure 2. The remote host node has a function as the sender and connected to the Packet Gateway (PGW) node in point to point mode. Datarate between PGW and Remote host is 100Gbps. MME has role to control signaling session, PGW was connected to S-GW before eNodeB to send radio transmission using the LTE EPC core network which indicates that the network to be simulated is a non-standalone 5G network.

The parameters and its description are shown in Table 1. The scenario in this research is to change node density simulated by different scheduling algorithms. Changes in the number of nodes in the scenario vary from twenty to one hundred with intervals of 20 UEs. The scheduler algorithm that will be used is round robin and proportional fair. UEs positions are arranged randomly with constant position mobility model. The simulations generate traffics for voice and video by remote hosts. The number of packet size and data rate used in the simulation is adjusted to the characteristics of the packet size and data rate on one of the VoIP codecs G.729 and H.264 video codec. For G.729 voice codec, which has 8 Kbps data rate and 20 bytes packet size. H.264 video codec which has 386 Kbps data rate and 240 bytes packet size.

4 Result and analysis

After simulating voice and video traffic from the 5G mmWave network in NS3, we obtained performance results such as throughput, delay and fairness index, then be analyzed. The analysis divided into two parts for voice and video traffic to find out which better scheduler for the two services.

4.1 Simulation result for voice traffic

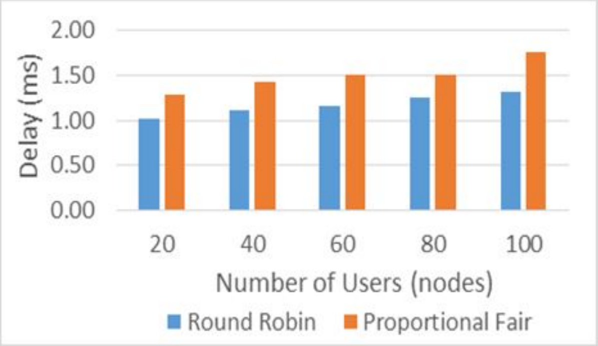


Figure 3: Delay on change number of users for voice traffic

Figure 3 shows the effect when increasing the number of users to the delay obtained from the voice traffic simulation. The lowest delay in round robin occurred on 20 UE with 1.023 ms, for proportional fair lowest delay occurred on 20 UE with 1.285 ms. On 100 UE, round robin and proportional fair generating the highest delay with 1.321 ms and 1.755 ms. Average delay obtained from round robin is 1.215 ms. This is 18.29 % lower than proportional fair with average delay of 1.487 ms. Based on figure 5, it can be conclude that delay for both scheduler increase, due to increase of number of UE make waiting time for each users to be served is getting longer. Round Robin has a better delay because for small packages, users queuing don't to take long time, different with Proportional fair which must take consider the channel quality.

Figure 4 shows effect the when increasing the number of users to the throughput obtained from the voice simulation. It show that, round robin has higher throughput than proportional fair. Round robin gets average throughput of 0.137 Mbps. This is 3.65 % better than proportional fair with average throughput of 0.132 Mbps. The lowest throughput in proportional fair occurred on 100 UE with 0.131 Mbps, and for round robin occurred on 100 UE with 0.132 Mbps. It show that the increase number of users, throughput decreased due to the bandwidth capacity will be shared with all users. Round robin has a higher throughput because this algorithm not consider the channel condition and has main purpose to balance between throughput and fairness among all the users.

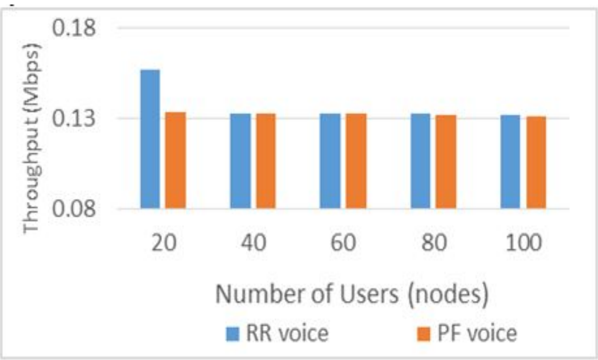


Figure 4: Throughput on change number of users for voice traffic

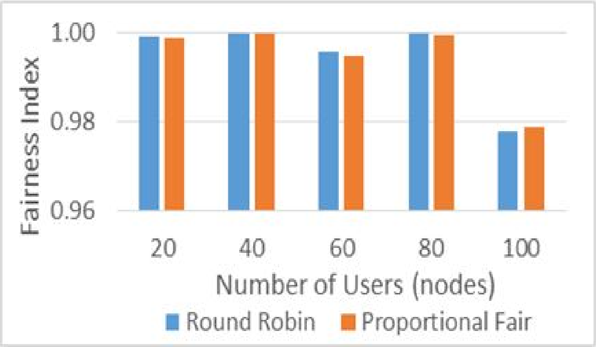


Figure 5: Fairness index on change number of users for voice traffic

Figure 5 shows the effect when increasing the number of users to fairness index in each scheduling algorithm for voice simulation. The average of fairness index obtained from the simulation of adding the number of users to the proportional fair algorithm is 0.994. It is lower than the round robin's fairness index that has 0.995. The value obtained by the round robin algorithm is greater because this algorithm does not consider channel conditions so that it offers a higher fairness value. Round robin and proportional fair have a decreasing fairness index value against the increase in the number of users. Its happened because of the increasing number of users, more users were served and reduce the value of fairness. Both schedulers show fairness due to the fairness index close to 1.

4.2 Simulation result for video traffic

Figure 6 shows the effect when increasing the number of users to the delay obtained from the video traffic simulation. Average delay obtained from round robin simulation with rising the UE is 3.105 ms. This is 2.19 % higher than proportional fair average delay with 3.037 ms which make proportional fair has better delay on video traffic. Round robin has higher delay because round robin not consider the channel condition, that make the delay for video traffic. Users served in sequence obtain longer delay.

Figure 7 shows the effect of increasing the users to the throughput obtained from the video traffic simulation. For proportional fair, the highest throughput occurred by 20 users with 2.869 Mbps. The highest throughput in round robin occurred on 20 users with 2.811 Mbps. It shown that round robin and proprotional fair has a decreasing throughput value towards increasing number of user. The average throughput of proportional fair obtained from the simulation is 2.820 Mbps. This is 1.24 % higher than round robin with 2.785 Mbps.

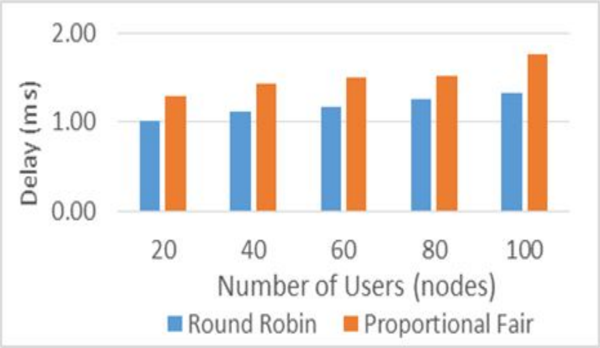


Figure 6: Delay on change number of users for video traffic

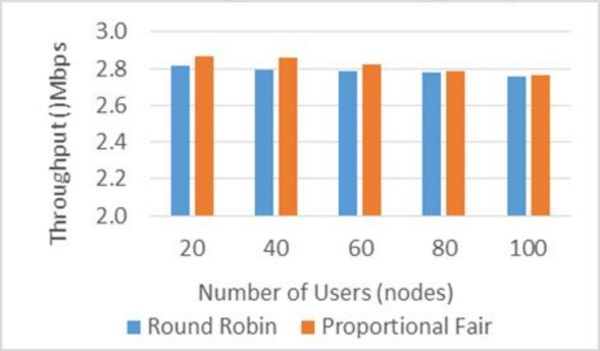


Figure 7: Throughput on change number of users for video traffic

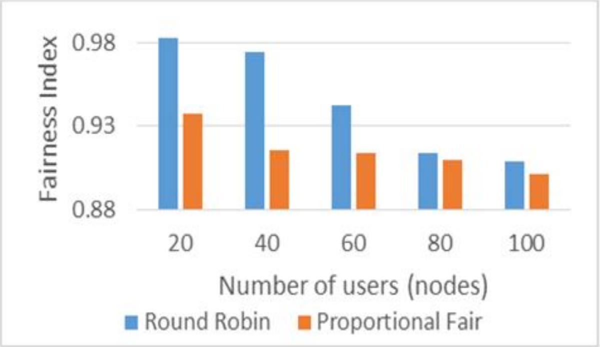


Figure 8: Fairness index on change number of users for video traffic

Figure 8 shows the effect when changing the number of users to fairness index in each scheduling algorithm for video traffic simulation. The average of fairness index obtained from the simulation of adding the number of users to the round robin algorithm is 0.94. It is 3.19 % better than proportional fair's fairness index that has 0.91. Round robin and proportional fair have a decreasing fairness index value against the increase in the number of users. Its happened because of the increasing number of users, more users were served and reduce the fairness. Round Robin has higher fairness index than proportional fair because round robin not consider channel condition and prioritize fairness among users.

5 Conclusion

In this paper, our work focuses on scheduling in a 5G network with a new MAC layer structure that has been proposed in previous studies. Based from simulation result, the choice of scheduling algorithm has affect on network performance. Proportional fair is better than round robin for throughput on video traffic with similar value of fairness index. Round robin has 3.19% better fairness index than proportional fairness and 2.19% higher average delay than proportional fair. It can be conclude that proportional fair is the right algorithm for video traffic. Round robin is right choice for voice traffic due to has better result for fairness and throughput. For further research, it is expected that more scheduler algorithms can be implemented in the new design of mac layer on the 5G network.

Bibliography

[1] Angri, I.; Mahfoudi, M.; Najid, A.; Bekkali, M. (2018). Exponential MLWDF (EXP-MLWDF) Downlink Scheduling Algorithm Evaluated in LTE for High Mobility and Dense Area Scenario, *International Journal of Electrical and Computer Engineering (IJECE)*, 8(3), 1618–1628, 2018.

[2] Aramide, S.O.; Barakat, B.; Wang, Y. et al. (2017). Generalized proportional fair (GPF) scheduler for LTE-A *2017 9th Computer Science and Electronic Engineering (CEEC).*

[3] Benitez-Perez, H.; Ortega-Arjona, J. ; Esquivel-Flores, O. et al. (2016). A Fuzzy Networked Control System Following Frequency Transmission Strategy, *International Journal of Computers Communication* & *Control*, 11(1), 11–25, 2016.

[4] Boccardi, F.; Heath, R. W.; Lozano, A. et al. (2014). Five disruptive technology directions for 5G, *IEEE Communications Magazine*, 52(2), 74–80, 2014.

[5] Capozzi, F.; Piro, G.; Grieco, L. A. et al. (2013). Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey, *IEEE Communications Surveys & Tutorials*, 15(2), 678–700, 2013.

[6] Carpin, M.; Zanella, A.; Rasool, J. et al. (2015). A performance comparison of LTE downlink scheduling algorithms in time and frequency domains, *2015 IEEE International Conference on Communications (ICC)*, London, 15(4), 3173–3179, 2015.

[7] Donoso, Y.; Lozano-Garzon, C.; Camelo, M. et al. (2014). A Fairness Load Balancing Algorithm in HWN Using a Multihoming Strategy, *International Journal of Computers Communication* & *Control*, 9(5), 555–569, 2014.

[8] Dutta, S.; Mezzavilla, M.; Ford, R. et al. (2017). Frame Structure Design and Analysis for Millimeter Wave Cellular Systems, *Wireless Communications IEEE Transactions* , 16(3), 1508–1522, 2017.

[9] Ford, R.; Zhang, M.; Mezzavilla, M. et al. (2017). Achieving Ultra-Low Latency in 5G Millimeter Wave Cellular Networks, *IEEE Communications Magazine*, 55(3), 196–203, 2017.

[10] Gomez, K.; Goratti, L.; Granelli F. et al. (2014). A comparative study of scheduling disciplines in 5G systems for emergency communications, *1st International Conference on 5G for Ubiquitous Connectivity*, 40–45, 2014.

[11] Hassebo, A.; Muath Obaidat, M.; Ali M. A. (2018). Commercial 4G LTE cellular net works for supporting emerging IoT applications, *2018 Advances in Science and Engineering Technology International Conferences (ASET)*, IEEE, 1–6,2018.

[12] Ismail, S.; Ali, D. M.; Yosuf, A.L. (2019). MECC scheduling algorithm in vehicular environment for uplink transmission in LTE networks, *International Journal of Electrical and Computer Engineering (IJECE)*, 9(2), 1191–1200, 2019.

[13] Jha, M.; Prateek, K.; Jaiswal, N. et al. (2016). Comparative Analysis of MAC Scheduling Algorithms in Long Term Evolution Networks using NS3, *Asian Journal of Enginnering Technology and Innovation*, 4(7), 124–127, 2016.

[14] Kawser, M. T.; Farid, H.M.A.B.; Hasin, A.R. et al. (2012). Performance Comparison between Round Robin and Proportional Fair Scheduling Methods for LTE, *International Journal of Information and Electronics Engineering (IJIEE)*, 2(5), 2012.

[15] Marwan, A.A.; Perdana, D.; Sanjoyou, D.D. (2019). Performance Analysis of RAW Impact on IEEE 802.11ah Standard Affected by Doppler Effect, *International Journal of Computers Communications & Control*, 14(2), 212–219, 2019.

[16] Mezzavilla, M.; Zhang, M.; Polese, M. et al. (2018). End-to-End Simulation of 5G mmWave Networks, *Communications Surveys & Tutorials IEEE* , 20(3), 2237–2263, 2018.

[17] Niu, Y.; Li, Y.; Jin, D et al.(2015). A Survey of Millimeter Wave (mmWave) Communications for 5G: Opportunities and Challenges, . *CoRR*, abs/1502.07228.

[18] Perdana, D.; Dewanta,F.; Wibawa, I.P.D. (2017). Extending Monitoring Area of Production Plant Using Synchronized Relay Node Message Scheduling, 2017 Inter national, *ournal of Communication Networks and Information Security*, 20(3), 2237–2263, 2018.

[19] Putra, M.A.P.; Perdana D.; Negara, R.M. (2017). Performance Analysis of Data Traffic Offload Scheme on Long Term Evolution (LTE) and IEEE 802.11AH, *Telekomnika*, 15(4), 1659–1665, 2017.

[20] Rangan, S.; Rappaport, T. S.; Erkip, E. (2014). Millimeter-Wave Cellular Wireless Networks: Potentials and Challenges, *Proceedings of the IEEE*, 102, 366–385, 2014.

[21] Wulandari, T.; Perdana D.; Negara, R.M. (2018). Node Density Performance Analysis on IEEE 802.11ah Standard for VoIP Service, *International Journal of Communication Networks and Information Security*, 10(1), 2018.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 540-556, August 2019.

Top-N Recommendation based on Mutual Trust and Influence

D.W. Seng, J.X. Liu, X.F. Zhang, J. Chen, X.J. Fang

Dewen Seng*
School of Computer Science and Technology, Hangzhou Dianzi University
Hangzhou 310018, China
*Corresponding author: sengdw@163.com

Jiaxin Liu
School of Computer Science and Technology, Hangzhou Dianzi University
Hangzhou 310018, China

Xuefeng Zhang
School of Computer Science and Technology, Hangzhou Dianzi University
Hangzhou 310018, China

Jing Chen
School of Computer Science and Technology, Hangzhou Dianzi University
Hangzhou 310018, China

Xujian Fang
School of Computer Science and Technology, Hangzhou Dianzi University
Hangzhou 310018, China

Abstract: To improve recommendation quality, the existing trust-based recommendation methods often directly use the binary trust relationship of social networks, and rarely consider the difference and potential influence of trust strength among users. To make up for the gap, this paper puts forward a hybrid top-N recommendation algorithm that combines mutual trust and influence. Firstly, a new trust measurement method was developed based on dynamic weight, considering the difference of trust strength between users. Secondly, a new mutual influence measurement model was designed based on trust relationship, in light of the social network topology. Finally, two hybrid recommendation algorithms, denoted as FSTA(Factored Similarity model with Trust Approach) and FSTI(Factored similarity models with trust and influence), were presented to solve the data sparsity and binarity. The two algorithms integrate user similarity, item similarity, mutual trust and mutual influence. Our approach was compared with several other recommendation algorithms on three standard datasets: FilmTrust, Epinions and Ciao. The experimental results proved the high efficiency of our approach.

Keywords: mutual trust, mutual influence, social recommendation system, cold start, data sparsity.

1 Introduction

The continuous expansion of e-commerce provides users with direct access to a growing number and variety of products. In this case, the users need to spend a lot of time looking for their desired products. This poses a huge challenge to information consumers and producers. To cope with the challenge, it is necessary to recommend the information and products favored by the users according to their interest features and purchase behavior, that is, to set up a recommendation system. For the users, the recommendation system enables them to pinpoint desired information out of a massive amount of data; for the merchants, the system helps to decide on which products to sell to a group of users and provide satisfactory services that enhance user loyalty [14].

Collaborative filtering recommendation is an important way of personalized recommendation, thanks to its ability to discover new interest points of users and a dazzling array of resources [4]. However, this recommendation approach is bottlenecked by problems like cold start and data sparsity in its algorithms. How to alleviate the two problems with the social attributes of the users has become the main task in the design of the recommendation system [17]. Among the possible solutions, the recommendation algorithm combined with social information, i.e., the social recommendation algorithm, comes as an effective means to solve the said problems [14]. The social information main includes the mutual trust and influence between users, both of which are essential to social activities [15].

The mutual trust has long been recognized as the key to recommendation system [1–3, 8–10, 12, 14, 15, 23]. In such a system, the rating can be predicted more accurately by replacing or supplementing the rate-based similarity [3]. The trust-based recommendation can effectively suppress the cold start problem and improve the accuracy and coverage of the recommendation [16]. In fact, trust communication has been proved as a social science, the key to social network analysis and an important phenomenon in recommendation scenarios [10]. But the existing measure of trust is only represented by 0 and 1 binary [8,9], and there is no clear value to indicate the specific trust between users. Hence, This paper not only improves the recommendation effect, but also alleviates the sparseness of social network data through the specific measurement of trust between users. In addition, through the study of social networks, we have found a new inspiration, what is the trust in social networks? In the traditional recommendation system, few people combine mutual trust and mutual influence. But in social networks, user influence really exists and has a certain impact on the recommendations between users [15].

In light of the above analysis, this paper attempts to find out the social relationship data that guide the recommendation, disclose and weight the effect of the guidance on user preference, and improve the results of the social recommendation system. For these purposes, two new hybrid recommendation algorithms were proposed for top-N recommendation, based on the factorized similarity model based on trust (FST). The two algorithms are respectively denoted as FSTA and FSTI. The innovations of our research are summed up as follows:

(1) Based on dynamically weighted trust relationship, a new mutual trust measurement method was put forward according to the difference in mutual trust strength. The method takes account of the direct and indirect trust relationships between users, thus improving the recommendation accuracy.

(2) Using the topology of social network, a new mutual influence measurement model was developed based on trust relationship. Considering both direct and indirect mutual influences, the proposed model makes full use of the implicit information in trust relationship.

(3) Two new hybrid top-N recommendation models, involving user similarity, item similarity, mutual trust and mutual influence, were designed to solve the data binarity and sparsity of mutual trust, and used to explore the existence of social network users, identify potential trust relationships, and set up a mutual trust network.

(4) Our models were proved efficient through repeated comparative verifications on three standard datasets, namely, FilmTrust, Epinions and Ciao.

2 Literature review

Social recommendation refers to the construction of a social relationship network between users based on the additional input of their social relationship information [20]. As long as the network users have direct or indirect social relationships, it is possible to generate proper recommendations for new users based on the interest models of the known users [17]. A collaborative filtering algorithm that combines social relationships is called a social recommendation algorithm. Many scholars and researchers have probed deep into the theories, methods and applications of social recommendation systems [10, 20, 23, 24]. Below is a brief review of some representative studies.

To improve the accuracy of rating prediction, Fang et al. [3] replaced and supplemented rate-based similarity in recommendation systems with trust. Wang and Ma [18] considered and weighted user's rating and trust similarity. Moradi and Ahmadian [12] proposed a trust-based singular value decomposition technique, TrustSVD, which evaluates both the explicit and implicit effects of trust. The above studies show that the inclusion of trust can enhance the accuracy of the recommendation system. On this basis, it is assumed that the recommendation accuracy can be improved through the combination of trust and other important social factors.

The advent of social networks has shortened the distance and enhanced the links between people, providing an excellent experimental platform and a huge amount of data for impact research [8]. Taking this golden opportunity, many researchers have analyzed and simulated the mutual influence in many social networks, and achieved fruitful results with huge application values [21, 25]. For instance, Li et al. [15] developed social recommendation algorithms through collaborative filtering, based on mutual trust and influence. With its important impact on user behavior in social networks, the mutual influence should be fully considered in the identification of influential users, such as to enhance the performance and accuracy of the recommendation system.

The personalized top-N recommendation system has a direct bearing on many real-world applications, such as e-commerce platforms and social networks [22]. However, the existing top-N recommendation methods become less effective with the growth in data sparsity. This problem can be solved by introducing social information like mutual trust and preference to the recommendations system. In this way, it is possible to mine out the implicit information from the social network, and overcome the defect of sparse datasets.

3 Recommendation based on mutual trust and influence

This section introduces mutual trust and influence to the social recommendation method of the FST, aiming to improve the effectiveness of traditional collaborative filtering. The two-module structure of the FST model combining the mutual trust and influence is shown in Figure 1, where the blue dashed lines indicate that a trusted user has an influence on the truster, and the black dotted lines indicate user u's rating of item j. In the left half of the model, user u trusts and is influenced by user v, who has rated item j as $r_(v, j)$; the goal is to predict user u's rating $r_(u, j)$ for item j, considering the mutual trust and influence between the two users. In the right half of the model, user k trusts and is influenced by user u, and has rated item j as $r_(k, j)$; the rating may affect user u's rating $r_(u, j)$ for the same item j.

Following this model, the social network relationship and a user's item ratings are analyzed to predict the item ratings of the other user. Finally, the item set with the higher rating is recommended to the users. Let $U = u_1, u_2, \cdots, u_m$, $I = i_1, i_2, \cdots, i_n$ and $R = [r_{u,i}]_{m \times n}$ be the set of m users, the set of n items, and the matrix of item ratings in the rating-based collaborative filtering algorithm, where $r_{u,i}$ is user u's rating of item i (the rating is generally a real number

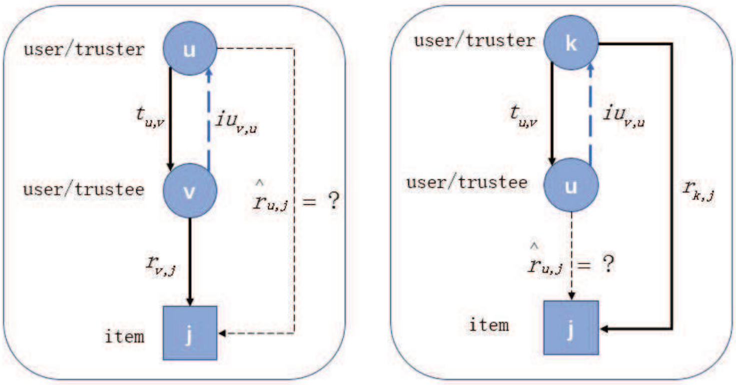


Figure 1: FST model combining mutual trust and influence

between [0, 5], and positively correlated with the user's interest in the item). The set of items rated by user u and the set of users having rated item i are respectively expressed as $I_u = i|r_{u,i} \neq 0$ and $U_i = u|r_{u,i} \neq 0$.

In a social network, the users' mutual trust can be represented by $T = [t_{u,v}]_{m \times m}$, where $t_{u,v}$ is the trust between u and v; the users' mutual influence can be represented by $IU = [iu_{u,v}]_{m \times m}$, where $iu_{u,v}$ is the influence between u and v. Note that each user in the social network has N friends.

3.1 Mutual trust ranking

If taken as the auxiliary information, mutual trust, an important type of social information, can effectively solve data sparsity problem in collaborative recommendation algorithms. In real-world applications, however, there following problems still exists: (1) The trust relationship is too sparse to be measured accurately; (2) The trust relationship is often expressed in the binary form (0 or 1), failing to weigh the exact trust strength between users.

Dynamically weighted trust measurement model

This subsection explores the rating prediction based on user's social relationship and constructs a model for mutual trust measurement. To begin with, the two users, A and B, in the social network are assumed to have only one interaction, i.e. rated the same item. During the interaction, each user has an impact on the other's trust if both of them have given the same rating. The success of the interaction hinges on the rating difference between the two users. The interaction process is shown in Figure 2, where the uppercase letters in circular boxes are items and the lowercase letters in rectangular boxes are users. It can be seen that, both users A and B have rated the same item a, i.e. engaged in an interaction. If the two users' ratings on the item is not above the threshold ϵ, then the interaction is successful; otherwise, the interaction is a failure.

Mutual trust is the accumulation of subjective feelings of individuals on each other. For two users u and v, the greater user u trusts user v, the more its interactions with v. The initial mutual trust $Init(u, v)$ can be defined as:

$$Init(u,v) = \frac{min(I_u \bigcap I_v, D)}{D} \tag{1}$$

where $I_u \cap I_v$ is the number of interactions between the two users, i.e. the number of items rated the same by the two users; D is an adjustable threshold that measures the minimum

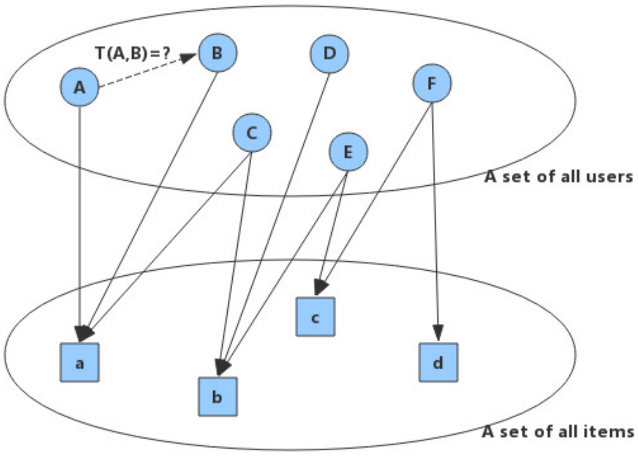


Figure 2: The interaction between users

number of interactions when two users fully trust each other. If $I_u \cap I_v$ is not greater than D, then the effective weight will come into force; otherwise, the initial strength of the mutual trust will be set to 1. Since the users differ in the number of ratings, the criterion for full trust may vary with users. Thus, the threshold for each user can be set to $D_u = \sqrt{|I_u|}$.

Next, the preference of user A for item c can be defined as [19]:

$$P(A,c) = \frac{\sum_m \in A_c sim(A,m)}{|A_c|} \tag{2}$$

where, A_c is to the set of m users having rated item c. Equation (2) shows that the preference of the users in set A_c for item c increases with their similarity with user A.

Since each pair of users will optimize the mutual trust through interaction, the trust between the two users $T(U,V)$ can be calculated by assigning a difference weight to the item based on the difference in the two users' preferences for the same item:

$$T(A,B) = \begin{cases} IniTD(A,B)\frac{\sum_{c\in success} P(A,c) - \sum_{c\in failure} P(A,c)}{\sum_{c\in success} P(A,c) + \sum_{c\in failure} P(A,c)} & ,T(A,B) \\ 0 & ,Otherwise \end{cases} \tag{3}$$

where, $Init(u,v)$, the initial trust between the two users is a dynamic weighting factor. Obviously, the fewer commonly rated items, the smaller the numerator, and the less the initial trust will contribute to the final mutual trust. The trust of each user to the other users can be computed by equation (3), and a set of trusted users can be obtained by filtering the results against the threshold.

FSTA model

The rating $r_{u,i}$ of user u on item i was predicted based on mutual trust, item similarity and user similarity. The mutual trusts of each target user with its trustee(s) and truster(s) were obtained by equation (3). After the calculation, the list of trust users was updated, and the implicit information in the user rating matrix was obtained to obtain the directed trust between users. The item similarity was the inner product of two lower-order matrices X and Y, where $X \in R^{n\times d}$ and $Y \in R^{n\times d}$. Note that $d << n$ is the number of potential feature vectors associated with the item. The user similarity was obtained from two lower-order matrices $P \in R^{m\times d}$ and $Q \in R^{m\times d}$, where $d << m$ is the number of potential feature vectors associated with the user.

Then, the ratings were ranked to produce a set of recommended items for each target user. The ratings of the user can be predicted as:

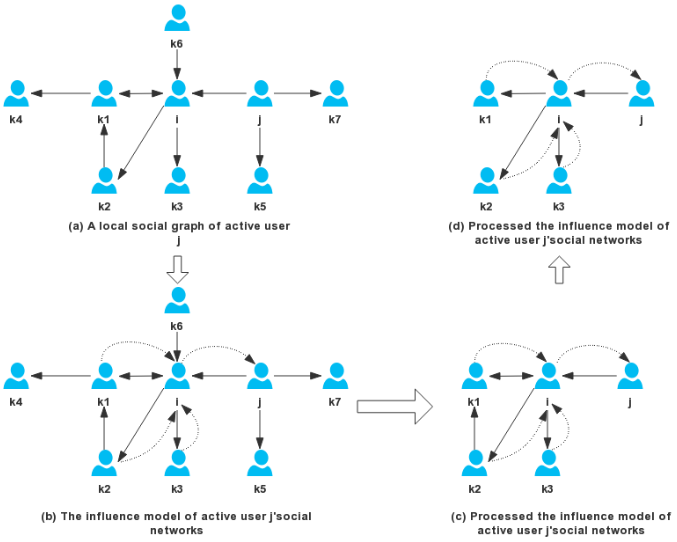
$$\begin{aligned}\widehat{r}_{u,i} = b_i + s|U_{i-u}|^{-\beta} \sum_{v\in U_{i-u}} p_v^T q_u + (1-s)|I_{u-i}|^{-\alpha} \sum_{j\in I_{u-i}} x_j^T y_j \\ + \sigma|T_u^+|^{-z} \sum_{a\in T_u^+} p_a^T y_i + (1-\sigma)|T_u^-|^{-\chi} \sum_{b\in T_u^-} p_b^T y_i\end{aligned} \tag{4}$$

where, α, β, z and $\chi \geq 0$ are the number of rated items, that of similar users, that of trustees and that of trusters; b_i is the bias of item i; s and $\delta \in [0, 1]$ are user similarity and the relative importance of trust, respectively. For each pair of trustee $a \in T_u^+$ and truster $b \in T_u^-$, the inner products $p_a^T y_i$ and $p_b^T y_i$ describe the influences of users a and b on the trust of the target user, respectively.

3.2 Mutual influence ranking

By mining the information in the trust relationship, the author put forward a method to calculate the implicit influence between users, taking account of both the direct and indirect trust relationships between users.

Influence calculation



(a) A local social graph of active user j

(d) Processed the influence model of active user j'social networks

(b) The influence model of active user j'social networks

(c) Processed the influence model of active user j'social networks

Figure 3: Local social network

As shown in Figure 3, the local social network consists of 9 users, respectively denoted as i, j, $k1$, $k2$, $k3$, $k4$, $k5$, $k6$ and $k7$. The goal is to estimate the trust strength between users i and j. The trust strength cannot be determined accurately if only the set of trusts (solid lines in Figure 3(a)) is considered. To fully exploit the information in the trust relationship, a trust measurement model was designed based on dynamic weight. By definition, user j is affected by user i when it trusts that user. Then, the nodes with no influence on node j were removed. After that, the trust relationships with no influence were deleted from the network.

Inspired by the above research, an improved algorithm was proposed to identify the influence of users in the trust network.

Definition 1. In a mutual trust network G_T, the edge weight w_ij can be defined as the trust strength between users i and j, that is, $w(i,j) = T(i,j)$. The nodes and directed edges in the network respectively stand for users and strengths between users.

Definition 2. The weight of node $w(i)$ can be interpreted by the trustee of the mutual trust network as $w(i)=\sum_{j\in T_i^+} w_{ij}$.

Definition 3. The relative importance of node j in the eyes of its neighbor node i can be calculated by $p(i,j) = w_{i,j}/w(i)$. If node j or its neighbor node k are of high relative importance, then node i will invest more time and energy in node j. This reflects the fact that people often devote more time and energy to those with great importance or importance relationships.

Definition 4. In light of the features of mutual trust network, an improved model was designed to measure mutual influence $C(i)$:

$$C(i,j) = \begin{cases} p(i,j) + \sum\limits_{k\in T_i^+} p(i,j)p(i,k), C(i,j) \geq v_c \\ \hfill 0, Otherwise \end{cases} \tag{4}$$

where, $p(i,j)$ and $\sum p(i,j)p(i,k)$ are the direct and indirect influences of user i on user j, respectively. The latter is determined by the number of intermediate nodes k between node i and node j. It can be seen from equation (5) that the value of $C(i,j)$ is positively correlated with the number of trust neighbors of each node and the closeness between network nodes.

FSTI model

To predict each user's rating of an item, it is assumed that user u in the above FSTA model is affected by a set of users $IU_u = \{w|iu_{u,w}\}$. On this basis, another social recommendation algorithm FSTI was constructed based on the implicit influence of mutual trust.

Firstly, the item rating was modelled by matrix decomposition. The rating prediction formula can be derived from the results of equation (5) and the mutual influence matrix as:

$$\begin{aligned} \widehat{r}_{u,i} = b_i + s|U_{i-u}|^{-\beta} \sum_{v\in U_{i-u}} p_v^T q_u + (1-s)|I_{u-i}|^{-\alpha} \sum_{j\in I_{u-i}} x_j^T y_j \\ + \delta|T_u^+|^{-z} \sum_{a\in T_u^+} p_a^T y_i + (1-\delta)|T_u^-|^{-\chi} \sum_{b\in T_u^-} p_b^T y_i + |IU_u|^{-\theta} \sum_{w\in IU_u} p_w^T y_i \end{aligned} \tag{6}$$

where, $\theta \geq 0$ is the number of users that influences user u (hereinafter referred to as influential users); IU_u is the set of influential users; s and δ are the control parameters; $p_w^T y_i$ is the inner product equal to the influence of each influential user $w \in IU_u$ over user u.

Next, the variables b, P, Q, X and Y can be computed by the following objective function:

$$J = \frac{1}{2}\sum_{u\in C} \sum_{i\in I_u^+, j\in I_u^-} ||(r_{u,i} - r_{u,j}) - (\widehat{r}_{u,i} - \widehat{r}_{u,j})||^2 + \frac{\lambda}{2}(||P||_F^2 + ||Q||_F^2 + ||X||_F^2 + ||Y||_F^2 + ||b||_F^2) \tag{7}$$

where, C is the set of all users; I_u^+ is the set of items rated by user u; I_u^- is the set of items not rated by user u; $|| \cdot ||_F^2$ is the Frobenius norm.

For convenience, the regularization parameter λ was employed to process all the relevant variables. The recommendation effect can be greatly improved through proper adjustment and assignment of these variables.

Compared with the FST, the FSTI has the following features:

(1) The FSTI replaces the trust matrix T with the trust strength estimation matrix S, rather than assume that a user has the same trust strength with any friend;

(2) The FSTI estimates the mutual influence matrix IU by the implicit influence of trust relationship, revealing the implicit mutual influence;

(3) The FSTI improves the recommendation accuracy through the comprehensive consideration of item similarity, user similarity, mutual trust and mutual influence.

3.3 Model learning and complexity analysis

FSTA complexity

First, the gradients and update rules for all variables were calculated for model training (lines 3-44, Algorithm 1). All variables were initialized with a random value in $(0, 0.01)$ (line 1). In each iteration, the overall computing cost in Algorithm 1 was approximately $O(nb)$, where n is the number of ratings in the training set and b is the mean number of users evaluating the item, and the sampling factor ρ (line 5) was used to randomly sample a set of negative instances of Z to train the model. The variables were repeatedly updated by the stochastic gradient descent (SGD) rules (lines 16-44) until the loss value had converged or the maximum number of iterations had been reached. Next, the converged user potential feature matrix P and the item implicit feature matrix X were outputted, marking the end of the training (line 45). Finally, all learned variables were returned as outputs.

FSTI complexity

The FSTI follows basically the same training procedure as the FSTA. To save space, the details on the implementation of the algorithm are omitted here. The pseudocode of the FSTI is listed in Algorithm 2. The trained model was used to predict each user's ratings of unknown items, and the items with the highest ratings were collected into a recommendation set.

Summary of FSTA and FSTI algorithm complexity

Referring to Algorithm 1, the FSTA time complexity of our method mainly includes the evaluation of the objective function and the calculation of the gradient of each variable. For each iteration, the total computation time cost of Algorithm 1 is $O(n_t bd(|R| + |T|))$, where n_t is the number of training matrices, b represents the average number of graded items for the user, and d represents the feature vector dimension. Relatively speaking, FSTI increases the influence of influence on the basis of FSTA, so the time complexity of FSTI is $O(n_t bd(|R| + |T| + |IU|))$. Since the rating matrix R and the trust matrix T and the influence matrix IU are very sparse, the time complexity of our method FSTI is much lower than the matrix cardinality. The above analysis shows that our strategy for personalized recommendation successfully integrates user similarity, mutual trust and mutual influence.

Algorithm 1 The learning algorithm of FSTA

```
**Data:** $\alpha$, $\beta$, $z$, $\chi$, $\rho$, $\lambda$, $\eta$;
Initialize $b, X, Y, P, Q$ with random values in $(0, 0.01)$;
while $\zeta$ not converged do
    for all $u \in C$ do
        for all $i \in I_u^+$ do
            $Z \leftarrow sample(\rho, I_u^-)$
            $m_{ki} \leftarrow \sum_{k\in I_{u-j}} x_k$, $w_{ki} \leftarrow |I_{u-j}|^{-\alpha}$
            $m_{vi} \leftarrow \sum_{v\in I_{i-u}} p_v$, $w_{vi} \leftarrow |I_{i-u}|^{-\beta}$
            $m_a \leftarrow \sum_{a\in T_{\underset{u}{+}}} p_a$, $w_a \leftarrow |T_u^+|^{-z}$
            $m_b \leftarrow \sum_{b\in T_{\underset{u}{-}}} p_b$, $w_b \leftarrow |T_u^-|^{-\chi}$
            $g \leftarrow 0$, $h \leftarrow 0$, $t - out \leftarrow 0$, $t - in \leftarrow 0$
            for all $j \in Z$ do
                $m_{kj} \leftarrow \sum_{k\in I_{u-j}} x_k$, $w_{kj} \leftarrow |I_{u-j}|^{-\alpha}$
                $m_{vj} \leftarrow \sum_{k\in I_{j-u}} p_v$, $w_{vj} \leftarrow |I_{j-u}|^{-\beta}$
                $Compute\, \widehat{r}_{u,i}$, $\widehat{r}_{u,j}\, by\, Equation$
                $\widehat{r}_{u,j} \leftarrow 0$
                $e \leftarrow (r_{u,i} - r_{u,j}) - (\widehat{r}_{u,i} - \widehat{r}_{u,j})$
                $b_i \leftarrow b_i + \eta(e - \lambda b_i)$
                $b_j \leftarrow b_j - \eta(e - \lambda b_j)$
                $q_u \leftarrow q_u - \eta(e(w_{vj}m_{vj} - w_{vi}m_{vi}) + \lambda q_u)$
                $y_i \leftarrow y_i + \eta(e(w_{ki}m_{ki} + w_a m_a + w_b m_b) - \lambda y_i)$
                $y_j \leftarrow y_j - \eta(e(w_{kj}m_{kj} + w_a m_a + w_b m_b) - \lambda y_j)$
                $g \leftarrow g - e w_{ki} q_u$
                $h \leftarrow h + e(w_{kj} y_j - w_{ki} y_i)$
                $t - out \leftarrow t - out + e w_a (y_j - y_i)$
                $t - in \leftarrow t - in + e w_b (y_j - y_i)$
                for all $v \in U_{j-u}$ do
                    $p_v \leftarrow p_v - \eta(e w_{vj} q_u - \lambda p_v)$
                end for
            end for
            for all $v \in U_{i-u}$ do
                $p_v \leftarrow p_v - \eta(g/\rho + \lambda p_v)$
            end for
            for all $k \in U_{u-i}$ do
                $x_k \leftarrow x_k - \eta(h/\rho + \lambda x_k)$
            end for
            for all $a \in T_u^+$ do
                $p_a \leftarrow p_a - \eta(t - out/\rho + \lambda p_a)$
            end for
            for all $b \in T_u^-$ do
                $p_b \leftarrow p_b - \eta(t - in/\rho + \lambda p_b)$
            end for
        end for
    end for
end while
return $b$, $P$, $Q$, $X$, $Y$
```

Algorithm 2 The learning algorithm of FSTI

```
Data: $\alpha$, $\beta$, $z$, $\chi$, $\theta$, $\rho$, $\lambda$, $\eta$;
Initialize b, X, Y, P, Q with random values in(0,0.01);
while $\zeta$ not converged do
    for all $u \in C$ do
        for all $i \in I_u^+$ do
            $m_I \leftarrow \sum_{w \in IU_u} p_w$, $w_I \leftarrow |IU_u|^{-\theta}$
            $inf \leftarrow 0$
            for all $j \in Z$ do
                $y_i \leftarrow y_i + \eta(e(w_{ki}m_{ki} + w_a m_a + w_b m_b + w_I m_I) - \lambda y_i)$
                $y_j \leftarrow y_j - \eta(e(w_{kj}m_{kj} + w_a m_a + w_b m_b + w_I m_I) - \lambda y_j)$
                $inf \leftarrow inf + ew_I(y_j - y_i)$
            end for
            for all $w \in IU_u$ do
                $p_w \leftarrow p_w - \eta(inf/\rho + \lambda p_w)$
            end for
        end for
    end for
end while
return $b$, $P$, $Q$, $X$, $Y$
```

4 Verification and results

This section verifies the parameter robustness and sensitivity in our strategy through several experiments on three public datasets, and compares the effectiveness of our strategy with six top-N recommendation methods based on implicit feedbacks.

4.1 Experimental setup

Datasets

The experiment data include three datasets, namely, Flimtrust, Ciao, and Epinions [4]. Each dataset contains ratings and mutual trust of users in social network. Upon signup, each user was allowed to rate all items, and browse the ratings by other users to make more favorable decisions. In addition, the user can befriend any trusted user, forming a network of trust relationships. For the datasets Epinions and Ciao, the rating was given against a five-point scale, where 1 means strongly dislike and 5 means strongly like. The mutual trust was described in binary form: existence (1) and absence (0). Our approach was applied to the three datasets to estimate the potential mutual trust and make accurate recommendations. As shown in Table 1, all the three datasets are extremely sparse, except FilmTrust.

Table 1: Datasets

Data Set	FilmTrust	Ciao	Epinions
Users	1,508	7,375	40,163
Items	2,071	99,746	139,738
Ratings	35,479	278,483	664,824
Density	1.14%	0.0379%	0.0118%

Comparative methods

Our approach was compared with the following six top-N recommendation methods based on implicit feedbacks:

(1) BPR: Bayesian personalized ranking. Based on implicit feedback, the BPR is extended from the top-N ranking recommendation algorithm by the pairing hypothesis for item ordering, using implicit feedback, Matrix Factorization (MF) and K-Nearest Neighbors (KNN) model [19].

(2) GBPR: The BPR based on group preference. The GBPR is an improved version of the BPR with richer interaction between users [16].

(3) MostPop: The MostPop is the baseline method that ranks the ratings of an item by popularity, i.e. how frequently the item is rated or consumed by the user.

(4) FISM: Factored item similarity model. The FISM alleviates the sparsity of the existing top-N recommendation algorithm by taking the product of two low-dimensional latent factor matrices as the similarity matrix [11] .

(5) FST: Factored Similarity model with Trust+. The FST introduces the mutual trust matrix and user similarity matrix into the FISM, thereby alleviating the sparsity of existing top-N recommendation algorithm and enhancing the accuracy of ranking recommendation [9].

(6) FSTA, FSTI: It is our proposed method for comparing two methods. Among them, FSTI adds the influence of user influence on the basis of FSTA.

Evaluation indices

The previous studies [14,19] on recommendation systems have shown that the common error rate indices of rating prediction, such as the mean square error (MSE) and root mean square error (RMSE), cannot fully characterize the performance of the recommendation algorithm. In fact, the algorithm performance mostly depends on the recommendation of the top-N items. To fully verify the effect of our approach, the FSTI was combined with the BPR for top-N recommendation [19], and evaluated the hit rate by precision, recall rate and F1-measure:

$$P@N = \frac{1}{|U'|}\sum_{u\in U'}\frac{|R_N(u)\cap I'_u|}{N} R@N = \frac{1}{|U'|}\sum_{u\in U'}\frac{|R_N(u)\cap I'_u|}{I'_u} F1@N = \frac{2\cdot P@N\cdot R@N}{P@N + R@N} \tag{5}$$

where, I'_u is the set of items overestimated by user u; U' is the test set; $P@N$, $R@N$ and $F1@N$ are the precision, recall rate and $F1{-}measure$ under N recommended items, respectively. Sometimes, the $P@N$ and $R@N$ may contradict each other. In this case, the $F1 - measure$ should be taken into account to solve the contradiction. The greater the values of $P@N$ and $F1@N$, the better the recommendation effect.

The five-fold cross validation was adopted in our experiments. Each dataset was split randomly into 5 parts. In each iteration, 4 parts were allocated to the training set and the remaining one to the test set. The mean results of the 5 parts were taken as the final results. The number of recommended items was selected in 5, 10.

Note that common evaluation indices like mean absolute error (MAE) and RMSE are not applicable to our research, because they are mainly used to evaluate prediction performance rather than the top-N recommendation effect.

Parameter settings

The parameters in our experiments were configured empirically, e.g. the number of potential factors was set to $d = 10$. For the GBPR, the number of users in the set was fixed as 5 and the parameters were adjusted by the sampling factor $\rho \in [0, 1]$. For all BPR-based methods, the unsorted items for model learning were selected through uniform sampling [19]. For all decomposition-based models, parameters α, β, z, χ and θ were selected from a set of small values, i.e. $\{0.5, 1, 2\}$, and the sampling factor ρ was fixed at 10 (Algorithm 1), for the FISM satisfies the interval of $[11, 17]$. For all matrix-based methods, the setting of the regularization parameter λ was optimized in $\{0.000001, 0.00001, 0.0001, 0.01, 0.1\}$ by grid search.

4.2 Effect of parameters α, β, z, χ and θ

The five parameters α, β, z, χ and θ describes item similarity, user similarity, mutual trust and mutual influence. In our experiments, the value of each parameter is adjusted in the small set of $\{0.5, 1, 2\}$ when the control parameters and were 0.5. Tables 2 and 3 respectively list the parameter configurations corresponding to the five best P@5 results of the FSTA and the FSTI on each dataset. Obviously, the different parameter configurations led to different results, and the optimal parameters changed from dataset to dataset.

Table 2: The optimal parameter configurations of the FSTA on the three datasets.

Methods	Filmtrust				Ciao				Epinions			
Parameters	α	β	z	χ	α	β	χ	χ	α	β	χ	χ
1	2	0.5	0.5	0.5	0.5	2	0.5	0.5	0.5	1	0.5	1
	0.4192				0.02892				0.01222			
2	0.5	2	0.5	1	2	1	2	1	2	0.5	2	2
	0.4187				0.02843				0.01194			
3	0.5	2	2	0.5	2	0.5	1	2	2	2	0.5	0.5
	0.4186				0.02821				0.01177			
4	0.5	1	0.5	0.5	1	1	0.5	2	0.5	0.5	1	0.5
	0.4184				0.02808				0.01159			
5	0.5	1	0.5	1	1	2	2	0.5	1	0.5	1	2
	0.4184				0.02804				0.01158			

Table 3: The optimal parameter configurations of the FSTI on the three datasets.

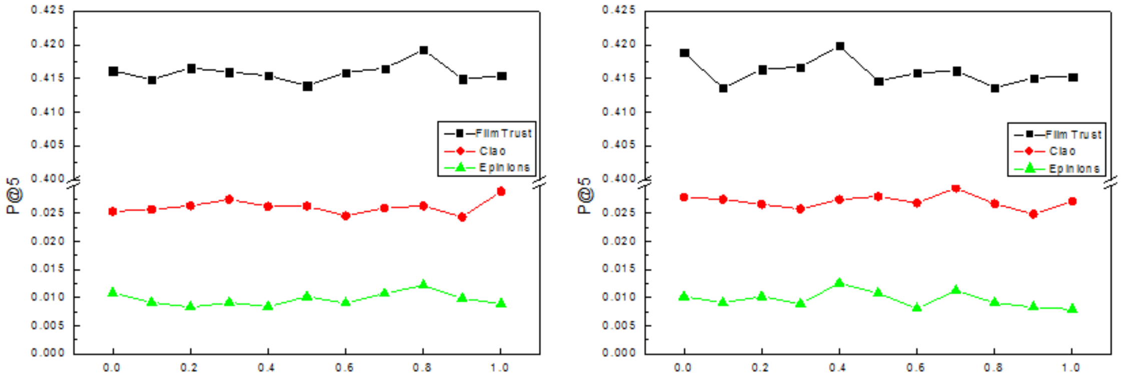
Methods	Filmtrust					Ciao					Epinions				
Parameters	α	β	z	χ	θ	α	β	z	χ	θ	α	β	z	χ	θ
1	0.5	0.5	2	1	2	0.5	1	2	1	0.5	1	2	2	0.5	1
	0.4198					0.02953					0.01260				
2	0.5	2	2	0.5	1	0.5	1	1	0.5	2	1	0.5	0.5	2	0.5
	0.4194					0.02941					0.01233				
3	1	2	1	2	0.5	2	1	0.5	1	0.5	0.5	0.5	2	1	2
	0.4192					0.02932					0.01215				
4	1	0.5	2	1	0.5	2	2	2	0.5	2	0.5	2	3	1	0.5
	0.4172					0.02930					0.01214				
5	0.5	1	0.5	1	2	2	2	1	2	2	1	2	1	1	0.5
	0.4192					0.02929					0.01206				

The experimental results of the FSTA corresponding to parameters α, β, z and $\chi$are recorded in Tables 4, and denoted by P@5. It can be seen that the results were optimal at $\alpha = 1$, $\beta = 2$

and $z > 0.5$. As a result, before making top-N recommendations, the item similarity, the trustee and the truster should be emphasized over user similarity. The experimental results of the FSTI corresponding to parameters α, β, z, χ and θ are recorded in Tables 5, and denoted by $P@5$. It can be seen that the results were optimal at $\alpha = 1$, $\beta = 2$ and $z > 0.5$ and $\theta < 2$. As a result, before making top-N recommendations, the item similarity, the trustee and the truster should be emphasized over user similarity.

4.3 Effect of control parameter s

For each dataset, the parameters α, β, z, χ and θ were set to their optimal values, the first control parameter δ was fixed at 0.5, and the second control parameter s was adjusted in the interval [0, 1] by a step size of 0.1. The effects of the adjustment on the FSTA and the FSTI are displayed in Figure 4(a) and Figure 4(b), respectively.



(a) FSTA (b) FSTI

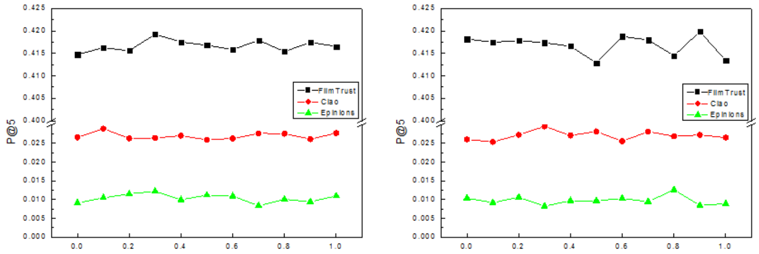
Figure 4: The effect of parameter s on our approaches FSTA and FSTI in terms of precision at 5

As shown in Figure 4, the optimal s values of the FSTA on FilmTrust, Ciao and Epinions were 0.8, 1 and 0.8, respectively, revealing that, in the FSTA model, user similarity has a stronger promotion effect on top-N recommendation accuracy than item similarity. Meanwhile, the optimal s values of the FSTI on FilmTrust, Ciao and Epinions were 0.4, 0.7, and 0.4, respectively. It can be seen that the weight of user similarity decreased in the FSTI model, while that of item similarity increased to a certain extent. Hence, both models will perform more excellently under proper parameter settings.

4.4 Effect of control parameter δ

For each dataset, the parameters α, β, z, χ and θ were set to their optimal values, the first control parameter s was fixed at 0.5, and the second control parameter δ was adjusted in the interval [0, 1] by a step size of 0.1. The effects of the adjustment on the FSTA and the FSTI are displayed in Figure 5(a) and Figure 5(b), respectively.

As shown in Figure 5, the optimal δ values of the FSTI on FilmTrust, Ciao and Epinions were 0.3, 0.1 and 0.3, respectively, revealing that, in the FSTI model, the truster is more important than the trustee of a user. Meanwhile, the optimal δ values of the FSTI on FilmTrust, Ciao and Epinions were 0.9, 0.3, and 0.8, respectively. It can be seen that the trusters of FilmTrust and Epinions are more important than those of Ciao.



(a) FSTA (b) FSTI

Figure 5: The effect of parameter δ on our approaches FSTA and FSTI in terms of precision at 5

4.5 Comparison with other methods

Table 4: The precisions of all algorithms on the three datasets (the precision of our approach is in bold).

Data	d	GBPR	MostPop	FISM	FST	FSTA	FSTI
FilmTrust	5	0.4124	0.4170	0.4171	0.4191	**0.4192**	**0.4198**
	10	0.3470	0.3503	0.3503	0.3514	**0.3525**	**0.3530**
Ciao ($\times 10^{-1}$)	5	0.2228	0.2677	0.2704	0.2714	**0.2892**	**0.2953**
	10	0.1827	0.2142	0.2141	0.2174	**0.2245**	**0.2337**
Epinions ($\times 10^{-1}$)	5	0.09353	0.1169	0.1147	0.1179	**0.1222**	**0.1260**
	10	0.07560	0.09171	0.09102	0.09187	**0.10410**	**0.10963**

Table 5: The F1-measures of all algorithms on the three datasets (the precision of our approach is in bold)

Data	d	GBPR	MostPop	FISM	FST	FSTA	FSTI
FilmTrust	5	0.4051	0.4095	0.4087	0.4099	**0.4103**	**0.4107**
	10	0.4458	0.4518	0.4516	0.4521	**0.4533**	**0.4534**
Ciao ($\times 10^{-1}$)	5	0.2063	0.2436	0.2495	0.2523	**0.2527**	**0.2841**
	10	0.2292	0.2662	0.2687	0.2720	**0.2743**	**0.2829**
Epinions ($\times 10^{-1}$)	5	0.1103	0.1298	0.1307	0.1330	**0.1419**	**0.1481**
	10	0.1111	0.1305	0.1315	0.1328	**0.1381**	**0.1467**

The $P@N$ and $F1@N$ of the FSTA and the FSTI were compared with those of the contrastive recommendation methods. The comparison results are listed in Tables 4 and 5. Overall, our approach outperformed the other methods under the same parameter configurations.

In the FilmTrust dataset, the MostPop achieved better results than the FISM. A possible reason lies in the consumption of the hot items in the dataset. In both Ciao and Epinions, the FISM outshined the MostPop, indicating the advantage of the factorized similarity model over the GBPR and BPR methods.

Moreover, the FSTA typically performed better than the methods, e.g. the FST, that integrates user similarity, item similarity and mutual trust. Note that the user rating matrix has a positive impact on the calculation of mutual trust, which comes from the trustee and the truster, respectively.

Finally, the comparison between the FSTA and the FST shows the influence of social impact on ranking performance. Our experimental parameters α, β, z, χ and θ were adjusted only in one group. Better results can be obtained by adjusting the parameter set.

5 Conclusion

Based on dynamically weighted trust relationship, a new mutual trust measurement method was put forward according to the difference in mutual trust strength. The method takes account of the direct and indirect trust relationships between users, thus improving the recommendation accuracy. Using the topology of social network, a new mutual influence measurement model was developed based on trust relationship. Considering both direct and indirect mutual influences, the proposed model makes full use of the implicit information in trust relationship.

Two new hybrid top-N recommendation models, involving user similarity, item similarity, mutual trust and mutual influence, were designed to solve the data binarity and sparsity of mutual trust, and used to explore the existence of social network users, identify potential trust relationships, and set up a mutual trust network. Our models were proved efficient through repeated comparative verifications on three standard datasets, namely, FilmTrust, Epinions and Ciao. The future research will explore the other factors that affect the mutual trust and further improve the performance of our models.

Acknowledgement

The work is supported by the National Science Foundation of China under Grant(No.61473108), and the National Experimental Teaching Demonstration Center of Computer Science in China. We thank Guibing Guo and his team for enlightening discussions.

Bibliography

[1] Callebert, L.; Lourdeaux, D.; BarthĂ¨s, J.P. (2018). Collective activity and autonomous characters: trust-based decision-making system, *Revue d'Intelligence Artificielle*, 31(1-2), 153-181, 2018.

[2] Coste, B.; Ray, C.; Coatrieux, G. (2017). Trust modelling and measurements for the security of information systems, *IngĂŠnierie des SystĂ¨mes d'Information*, 22(1), 19-41, 2017.

[3] Fang, H.; Bao, Y.; Zhang, J. (2014). Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation, *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 30-36, 2014.

[4] Guo, X.; Yin, S.; Zhang, Y.; Li, W.; He, Q. (2019). Cold start recommendation based on attribute-fused singular value decomposition, *IEEE Access*, 7, 11349-11359, 2019.

[5] Guo G.; Zhang J.; Yorke-Smith N (2016). A Novel Recommendation Model Regularized with User Trust and Item Ratings, *IEEE Transactions on Knowledge & Data Engineering*, 28(7), 1607-1620, 2016.

[6] Guo G.; Zhang J.; Zhu F. et al (2017). Factored similarity models with social trust for top-N item recommendation, *Knowledge-Based Systems*, 122, 17-25, 2017.

[7] Guo G(2019). List of Recommendation Data Sets, *https://www.librec.net/datasets.html*, 2011/6-2013/11.

[8] Han, Z.M.; Chen, Y.; Liu, W.; Yuan, B.H.; Li, M.Q.; Duan, D.G. (2017). Research on node influence analysis in social networks, *Journal of Software*, 28(1): 84-104, 2017.

[9] Jamali, M.; Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks, *ACM Conference on Recommender Systems*, 135-142, 2010.

[10] Kabbur, S.; Ning, X.; Karypis, G. (2013). Fism: factored item similarity models for top-n recommender systems, *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 659-667, 2013.

[11] Li, W.; Ye, Z.; Xin, M.; Jin, Q. (2017). Social recommendation based on trust and influence in SNS environments, *Multimedia Tools & Applications*, 76(9), 11585-11602, 2017.

[12] Moradi, P.; Ahmadian, S. (2015). A reliability-based recommendation method to improve trust-aware recommender systems, *Expert Systems with Applications*, 42(21), 7386-7398, 2015.

[13] Pan, W.; Chen, L. (2013). GBPR: group preference based Bayesian personalized ranking for one-class collaborative filtering, *Twenty-Third International Joint Conference on Artificial Intelligence*, 2691-2697.

[14] Pan, Y.; He, F.; Yu, H. (2018). Social recommendation algorithm using implicit similarity in trust, *Chinese Journal of Computers*, 41(1), 65-81, 2018.

[15] Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback, *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, 452-461, 2009.

[16] Tang, J.; Gao, H.; Liu, H.; Sarmas, A.D. (2012). eTrust: Understanding trust evolution in an online world, *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 253-261.

[17] Wu, M.X.; Dong, L.S.; Jie, Z.Y.; Hu, X. (2015). Research on social recommender systems, *Journal of Software*, (6), 1356-1372, 2015.

[18] Wang, M.; Ma, J. (2016). A novel recommendation approach based on users' weighted trust relations and the rating similarities, *Soft Computing*, 20(10), 3981-3990, 2016.

[19] Wang, Q.; Wang, J.H. (2015). Collaborative filtering recommendation algorithm combining trust mechanism with user preferences, *Computer Engineering and Applications*, 51(10), 261-265, 2015.

[20] Yang, X.; Guo, Y.; Liu, Y.; Steck, H. (2014). A survey of collaborative filtering based social recommender systems, *Computer Communications*, 41, 1-10, 2014.

[21] Yao, Q.; Shi, R.; Zhou, C.; Wang, P.; Guo, L. (2016). Topic-aware social influence minimization, *Proceedings of the 24th International Conference on World Wide Web*, 139-140 2015.

[22] Zhao, F.; Guo, Y. (2016). Improving Top-N recommendation with heterogeneous loss, *International Joint Conference on Artificial Intelligence*, 2378-2384, 2016.

[23] Zhao, H.Y.; Hou, J.D.; Chen, Q.K. (2015). Collaborative filtering recommendation algorithm combining time weight and trust relationship, *Application Research of Computers*, 32(12), 3565-3568, 2015.

[24] Zhang, D.; Sui, J.; Gong, Y. (2017). Large scale software test data generation based on collective constraint and weighted combination method, *Tehnicki Vjesnik*, 24(4), 1041-1050, 2017.

[25] Zhang, J.; Tang, J.; Li, J.; Liu, Y.; Xing, C.X. (2015). Who influenced you? Predicting retweet via social influence locality, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3), 25, 2015.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 557-573, August 2019.

Hyperparameter Importance Analysis based on N-RReliefF Algorithm

Y. Sun, H. Gong, Y. Li, D. Zhang

Yunlei Sun*
College of Computer & Communication Engineering
China University of Petroleum(East China), China
No.66, West Changjiang Road, Huangdao District, Qingdao 266580, China
*Corresponding author: sunyunlei@upc.edu.cn

Huiquan Gong
Faculty of Information Technology
Beijing University of Technology, China
No.100, Pingleyuan, Chaoyang District, Beijing, 100124, China
xinel_ghq@126.com

Yucong Li
College of Computer & Communication Engineering
China University of Petroleum(East China), China
No.66, West Changjiang Road, Huangdao District, Qingdao 266580, China
291731931@qq.com

Dalin Zhang
National Research Center of Railway Safety Assessment
Beijing Jiaotong University, China
No.3, Shangyuancun, Haidian District, Beijing, 100044, China
dalin@bjtu.edu.cn

Abstract: Hyperparameter selection has always been the key to machine learning. The Bayesian optimization algorithm has recently achieved great success, but it has certain constraints and limitations in selecting hyperparameters. In response to these constraints and limitations, this paper proposed the N-RReliefF algorithm, which can evaluate the importance of hyperparameters and the importance weights between hyperparameters. The N-RReliefF algorithm estimates the contribution of a single hyperparameter to the performance according to the influence degree of each hyperparameter on the performance and calculates the weight of importance between the hyperparameters according to the improved normalization formula. The N-RReliefF algorithm analyses the hyperparameter configuration and performance set generated by Bayesian optimization, and obtains the important hyperparameters in random forest algorithm and SVM algorithm. The experimental results verify the effectiveness of the N-RReliefF algorithm.

Keywords: Hyperparameter optimization, Bayesian optimization, RReliefF Algorithm.

1 Introduction

In the process of machine learning, the performance of the algorithm highly depends on the selection of hyperparameters, which has always been a crucial step in the process of machine learning. Automated machine learning, represented by Bayesian optimization algorithm, has recently achieved great success in hyperparameter optimization, which exceeds the performance of human experts in some cases.

However, there are some constraints and limitations in the selection of hyperparameters for Bayesian optimization algorithm. Researchers and users can only get the hyperparameter configuration after the operation of Bayesian optimization algorithm and cannot get the importance analysis of the hyperparameter configuration. Therefore, it is necessary to study the algorithm of hyperparameter importance analysis based on a wide range of data sets, so that researchers and users can understand which hyperparameter adjustment will significantly improve the performance of the algorithm.

This paper introduces the basic principle of Bayesian optimization algorithm and optimized random forest and SVM based on OpenML100 data set. By comparing with grid search and random search algorithm, it could be seen that the hyperparameter performance of Bayesian optimization algorithm is high and time-consuming. Therefore, we used the hyperparameter configuration and performance data generated by Bayesian optimization algorithm to analyze hyperparameter importance. Hyperparameter configuration based on the experimental data, we used N-RReliefF algorithm to evaluate the importance of interaction between hyperparameter, so as to determine the important hyperparameter in random forest algorithm and SVM algorithm.

2 Related works

Hyperparameter selection is a key step in machine learning process [12]. From the initial manual selection to automatic optimization using algorithms later, the evolving optimization algorithms have made great contributions to improving performance. Hyperparameter selection algorithms can be roughly divided into four categories: traditional algorithms (e.g., grid search algorithm [26], random search algorithm [2]), heuristic optimization algorithm [4,18,30], meta-learning algorithm [13,22], Bayesian optimization algorithm [14,28], etc.

The disadvantage of these hyperparameter selection algorithms is that they cannot provide researchers and users with information about the importance analysis of the selected hyperparameters and cannot understand the impact of different hyperparameters and their interactions on performance. Scientists have proposed a method for evaluating the importance of hyperparameter machine learning algorithms. Sequential parameter optimization (SPO) [1] is a model-based parameter optimization approach. SPO starts by running the target algorithm with parameter configurations. It then builds a response surface model based on Gaussian process regression and uses the models predictions and predictive uncertainties to determine the next parameter cofiguration to evaluate. In 2007, Nannen et al. [20] proposed an evolutionary algorithm for parameter correlation estimation. In 2009, Chiarandini et al. [6] used a linear mixed effect model to design and analyze the optimization algorithm. With a mixed-effects multi-linear regression they [8] assessed the individual and joint effect of problem features on the performance of both algorithms, within and across the instance classes defined by benchmark parameters. In [21] Probst formalized the problem of tuning from a statistical point of view, define data-based defaults and suggest general measures quantifying the tunability of hyperparameters of algorithms. Falkner proposed a new hyperparameter optimization method-BOHB [10], which combines the benefits of both Bayesian optimization and bandit-based methods, consistently outperforms both Bayesian optimization and Hyperband on a wide range of problem types. Breiman [29] uses random forests to assess the importance of attributes: If attributes are deleted from the data set, performance will be degraded, indicating that this attribute is important. Based on this principle, Forward Selection [15] predicts the performance of machine learning algorithms using a subset of hyperparameters, which is initialized to be empty and greedily chooses the next most important hyperparameter. Ablation Analysis [15] requires default settings and optimization settings, and calculates the ablation trajectory, which reflects the contribution of hyperparameters to the performance difference between the two settings.

3 N-RReliefF algorithm design

This paper introduces the basic principle of hyperparameter optimization of Bayesian optimization algorithm based on Gauss process modeling. To hyperparameter configuration and performance data generated by optimization process, we used N-RReliefF algorithm to evaluate the importance of hyperparameter, so that users can understand the important hyperparameter. As shown in Fig. 1, the input of hyperparameter optimization includes: algorithm A with configuration space, instance set and cost matrix C. The optimal hyperparameter configuration can be obtained by modeling and optimization of Bayesian optimization algorithm. At the same time, the trajectory of searching for the optimal hyperparameter configuration, as well as the hyperparameter configuration and its performance data can be obtained. Based on the output data, N-RReliefF algorithm is used to evaluate the importance of hyper-parameters and the effect of interaction between them on performance. Next, the two parts are explained in detail.

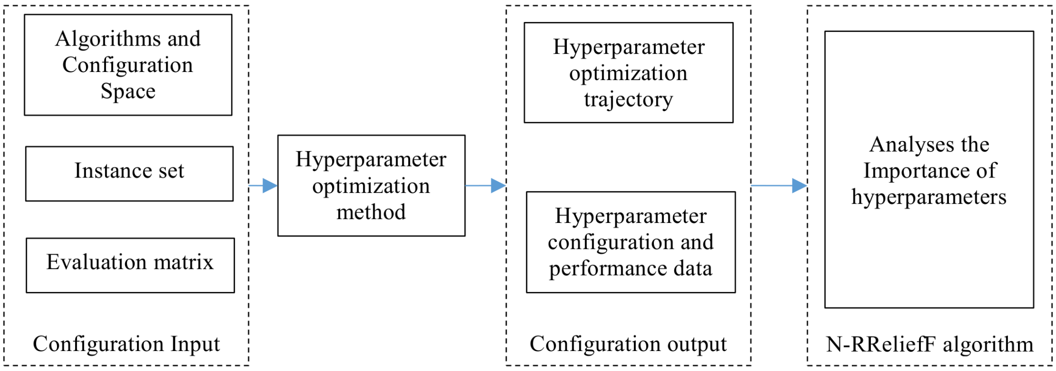

Algorithms and Configuration Space
Instance set
Evaluation matrix
Configuration Input
Hyperparameter optimization method
Hyperparameter optimization trajectory
Hyperparameter configuration and performance data
Configuration output
Analyses the Importance of hyperparameters
N-RReliefF algorithm

Figure 1: Algorithm configuration and analysis workflows

3.1 Bayesian optimization algorithm

Question definition

The hyperparameter selection problem of machine learning model is regarded as an unknown black box function optimization problem reflecting generalization performance. Let $\theta_1, ..., \theta_n$ represent n hyperparameters of machine learning algorithm, whose domain space is expressed as $\Theta_1, ..., \Theta_n$. The configuration space of the algorithm is defined as $\Theta = \Theta_1 \times ... \times \Theta_n$. The hyperparameters $\theta \in \Theta$ are trained on the training data set D_{train}, and the loss function $l(\theta, D_{train}, D_{valid})$ of machine learning algorithm is obtained on the verification set.

The objective function of hyperparametric combinations in optimization problems is defined as follows [11]:

$$f(\theta) = \frac{1}{k}\sum_{i=1}^{k} l(\theta, D_{train}^{(i)}, D_{valid}^{(i)}) \tag{1}$$

$$\theta^* = arg_{\theta\in\Theta} min f(\theta)$$

For an unknown objective function $f(\theta)$, Bayesian optimization algorithm searches for a hyperparameter configuration that minimizes the function $f(\theta)$ on a bounded set Θ. The basic idea of Bayesian optimization algorithm is to construct a probability model for function$f(\theta)$, establish evaluation criteria based on this model, determine the next hyperparameter for evaluation in configuration space Θ, and at the same time, all the information available in previous evaluation can be reused for learning the shape of objective function [9]. The use of historical data enables Bayesian optimization to find the minimum value of complex non-convex functions through fewer evaluations, but the corresponding cost is to perform more calculations to determine the next sampling point.

Therefore, the key of Bayesian optimization algorithm can be summarized as the following two parts [7]:

- Establishing a probabilistic model to evaluate the objective function instead of the original complex objective function, which is expensive to evaluate.
- The acquisition function is constructed by using the posterior information of the probability model to determine the next sampling point.

Question definition

There are many models to model the objective function, among which the Gauss process has been proved to be a convenient and powerful model optimization algorithm. Gauss process is a set of random variables. If these random variables obey Gauss distribution, then these random variables are Gauss process. A Gauss process consists of a mean function $m : \Theta \rightarrow R(m(\theta) = 0)$ and a covariance function $(kernelfunction) m : \Theta \rightarrow R(m(\theta) = 0)$.

Its concrete form is [25]:

$$f(\theta) \sim GP(m(\theta), k(\theta, \theta')) \tag{2}$$

Mean function $m(\theta) = E[f(\theta)]$, covariance function $k(\theta, \theta') = E[(f(\theta) - m(\theta)(f(\theta') - m(\theta')))]$, for simplicity, usually set mean function $m(\theta) = 0$. Covariance function is a function of calculating the similarity between two data points in Gauss process, which specifies the smoothness and amplitude of the unknown objective function. The selection of covariance function is very important, which affects the matching degree between Gauss process and data properties. This paper chooses Matérn $\frac{5}{2}$ Kernel Function [15]. Compared with other popular Gauss Kernel Functions, it has fewer constrained smoothness assumptions and is very helpful to the optimization settings.

The formulas are as follows:

$$k_{\frac{2}{5}}(\theta, \theta') = \theta_0(1 k_{\frac{2}{5}}(\theta, \theta') = \theta_0(1+\sqrt{5}d_\lambda(\theta, \theta')+\frac{3}{5}d_\lambda^2(\theta, \theta'))e^{-\sqrt{5}d_\lambda(\theta, \theta')}+\sqrt{5}d_\lambda(\theta, \theta')+\frac{3}{5}d_\lambda^2(\theta, \theta'))e^{-\sqrt{5}d_\lambda(\theta, \theta')} \tag{3}$$

Among them, θ_0 and λ denote the covariance amplitude and length dimensions respectively, and $d_\lambda(\theta, \theta') = (\theta, \theta')^T diag(\lambda)(\theta - \theta')$ denotes the Mahalanobis distance.

Given the input set $G = \theta_1, .., \theta_t$ and the output $y = f(\theta_1), f(\theta_2), ..., f(\theta_t)$ of the observation set, the Gauss process $GP(m, k)$ is adjusted. Due to the mixing of noise, the observed value is likely to be affected, and there is a certain deviation from the actual output value. In order to approach the actual situation, noise should be added to the probability distribution in the experiment.

The formula is as follows:

$$y = f(\theta) + \varepsilon \tag{4}$$

The noise ε satisfies the independent and identically distributed Gauss distribution: $p(\varepsilon) \sim N(0, \sigma^2)$. The prior distribution of y is $y \sim N(0, S + \sigma^2 I)$, I is n-dimensional unit matrix, and S represents the covariance matrix $k(\theta, \theta')$.

The joint prior distribution of the observed value y and predicted value $f(\theta_*)$ is as follows:

$$\begin{bmatrix} y \\ f(\theta_*) \end{bmatrix} \ N\left(0, \ \begin{bmatrix} S+\sigma^2 I & K_* \\ K_*^T & K_{**} \end{bmatrix}\right) \tag{5}$$

In the formula, θ_* represents the predictive input, $K_*^T = k(\theta_1, \theta_*), k(\theta_2, \theta_*), ..., k(\theta_t, \theta_*), K_{**} = k(\theta_*, \theta_*)$.

According to the estimation of posterior probability of input value by Gauss distribution, the predicted distribution at a given test point θ_* is expressed as [7]:

$$p(f(\theta_*)|G, y, \theta_*) = N(\overline{f(\theta_*)}, cov(f(\theta_*))) \tag{6}$$

Among,

$$\overline{f(\theta_*)} = K_*^T[S+\sigma^2 I] - 1y, cov(f(\theta_*)) = K ** - K * T[S+\sigma^2 I] - 1K* \tag{7}$$

GP evaluates $f(\theta_*)$ with all historical observation points as conditions, and then uses the posterior mean and variance of prediction to select the next set of superparameters on the basis of balanced development and exploration acquisition functions.

Acquisition function

This section introduces the active strategy of selecting the next evaluation point in Bayesian optimization: acquisition function, which is a function$\alpha : \chi \times \Theta \to R$ mapped from input space χ, observation space $\mathbb{R}$ and hyperparameter space Θ to real space.

The function is constructed from a posterior distribution obtained from known observation data sets $D_{1:t}$, and the next evaluation point θ_{t+1} is selected by maximizing its guidance:

$$\theta_{t+1} \in max_{x \in \chi} \alpha_t(\theta; D_{1:t}) \tag{8}$$

This paper uses the promotion-based (Expected Improvement, EI) strategy [19], which performs better. EI strategy has been proved to be effective in the evaluation of global optimization of many black box functions.

The closed form of EI strategy in Gauss process is as follows:

$$a_{EI}(\theta; D_{1:t}) = E[max(f_{min} - f(\theta), 0)] \tag{9}$$

f_{min} is the optimal solution based on observation set so far. Formula below describes the balance between the development and exploration of new sampling points. If the standard deviation of the prediction point is large, it means that the understanding of the point is small, and it is worth exploring; if the mean value is large, it means that the point may be the maximum point, which is worth developing. Because the initial sampling data is very few, the algorithm will sample the points with large standard deviation; when the sampling points increase, the standard deviation decreases, and the algorithm tends to the points with large sampling mean, and eventually converges to the global optimal value.

3.2 Importance assessment of hyperparameter

Bayesian optimization algorithm obtains the optimal hyperparameter configuration of machine learning algorithm through two important steps: GP process and iteration of acquisition function. However, due to the abstraction and black-box nature of its internal process, it is impossible to analyze the importance of hyperparameters. In order to increase the interpretability of the hyperparameters selected by Bayesian optimization algorithm and to understand the importance ranking of the hyper-parameters of the algorithm itself, an N-RReliefF algorithm is proposed to evaluate the importance of the hyperparameters.

Relief algorithm

The Relief algorithm [17] was originally used in the field of feature selection. The main idea of the Relief algorithm is to estimate the ability of this feature to distinguish adjacent samples according to the degree of discrimination of each attribute to an instance. Relief's process is to randomly select an instance I in the training set, search for k instances I_j which are similar to instance I, and the samples which belong to the same category with instance I in I_j are called H, and the samples of different categories are called M [9].

The weight $W[A]$ of attributes A are estimated according to the values of example I and H and M in I_j, and the approximate values of the probabilistic difference shown in formula below are obtained:

$$W\left[A\right]=P\left(diff.valueofA|nearestinst.fromdiff.class\right)-P\left(diff.valueofA|nearestinst.fromsameclass\right) \tag{10}$$

If instances I and H have different attribute values A, then attribute a separates two instances from the same kind of instances, and the formula is expressed as reducing the weight estimation $W[A]$. If instances I and M have different attribute values A, then attribute values a separates two instances from different instances in a formula that correspondingly increases the weight estimation $W[A]$. The bigger the weight of the feature is, the stronger the classification ability of the feature is; on the contrary, the weaker the classification ability of the feature is [27].

Among them, the difference of attribute value A between different instances I_1 and I_2 in $W[A]$ is defined as [27]:

$$dif(A,I_1,I_2)=0,value(A,I1)=value(A,I2)1,value(A,I1)\neq value(A,I2) \tag{11}$$

The above formulas can be further calculated as follows:

$$dif(A,I_1,I_2)=\frac{|value(A,I_1)-value(A,I_2)}{max(A)-min(A)} \tag{12}$$

N-RReliefF algorithm

The importance of hyperparameters of machine learning algorithm is evaluated. The input data used are the hyperparameter configuration and performance data of machine learning algorithm. On such data sets, performance data are continuous values, and can not be calculated using the latest samples of the same or different types presented in Relief. In order to solve this problem, RReliefF [23] introduces the probability of two different instances to determine whether two instances belong to the same class. The probability definition can simulate and predict the relative distance between two instances. RReliefF is currently mainly used in the field of feature selection. This section improves and fuses the RReliefF algorithm and proposes an N-RReliefF algorithm to evaluate the importance of the interaction between hyperparameters.

The main idea of the N-RReliefF algorithm for evaluating the importance of hyperparameters is to estimate the contribution of each hyperparameter to performance according to the degree of influence of each hyperparameter on performance. The N-RReliefF algorithm consists of two parts. The first part is to evaluate the importance of a single hyperparameter. In the training set, we randomly select a hyperparameter configuration instance I and select k instances I_j which are similar to the instance I. In order to judge whether the instance I_j and I belong to the same class, we introduce probability simulation and prediction of the relative distance between the two instances [24], as shown in formula below. Among them, θ denotes the probability of different hyperparameter values, P_{difA} denotes the probability of different categories in similar instances, P_{difC} denotes the probability of different categories in similar instances, and $P_{difC|difA}$ denotes the probability of different categories in similar instances with different hyperparameter values.

$$P_{difA} = P\left(difvalue\left(\theta\right)\left|similarinstance\right.\right) \;\; P_{difC} = P(difprediction|similar\ instance) \tag{13}$$

According to conditional probability:

$$P_{difC|difA} = P(difprediction|difvalue(\theta)\ similar\ instance) \tag{14}$$

Combined formula (11) is available:

$$W[\theta] = PdifC|difA \times PdifAPdifC - (1 - PdifC|difA) \times PdifA1 - PdifC \tag{15}$$

Repeat the above process k times to get $W[\theta]$, and evaluate the importance of a single hyperparameter according to the weight $W[\theta]$.

The second part is to measure the importance of hyper-parameters and understand the influence of the interaction between hyperparameters on the performance of machine learning algorithm. The N-RReliefF algorithm divide the contribution of the hyperparameters by the sum of all the contributions of the hyperparameters to normalization to calculates the importance of the hyperparameters.

The formula is defined as (17):

$$W[\theta_m \& \theta_n] = \frac{e^{W[\theta_m]+W[\theta_n]}}{e^{\sum W[\theta]}} \tag{16}$$

θ_m and θ_n denote two different hyperparameters, and $\sum W[\theta]$ denotes the sum of the importance weights of all hyperparameters.

This formula is a distortion of the normalization formula, which can more stably evaluate the influence of the interaction between hyperparameters on the performance. The whole process of the N-RReliefF algorithm is as follows: firstly, the contribution (weight) vector of each hyperparameter to the performance is calculated, the elements in the vector are accumulated, the importance of each hyperparameter is sorted by the accumulated value, and t significant hyperparameters are selected to enter the candidate subset of the hyperparameter, thus the iteration process begins. In the iteration process, the importance weights between the significant hyperparameters are calculated, and the importance weights between all the hyperparameters and the hyperparameters are finally output.

The flow chart of the algorithm is shown in algorithm 1.

In the algorithm, N_{dc}, $N_{dA}[\theta]$ and $N_{dC\&dA}[\theta]$ represent weight vectors of different predicted values (line 8), weight vectors of different attributes (line 10), and weight vectors of different predicted values and attributes (line 11). The algorithm calculates the importance weight $W[\theta]$ of each hyperparameter in line 16. H_{list} denotes the most important first t hyperparameters.

Variables $d(i, j)$ (lines 8, 10, 11) are used to measure the distance between two instances R_i and I_j. The basic principle is that closer instances should have greater impact:

$$d(i,j) = \frac{d_1(i,j)}{\sum_{l=1}^{k} d_1(i,l)} \;\; d_1(i,j) = e^{-(\frac{rank(R_i,I_j)}{\sigma})^2} \tag{17}$$

$rank(R_i, I_j)$ is the ranking of distance between instance I_j and instance R_i. σ is used to control distance, which is customized by users. Because the expected results can be interpreted by probability, divide the contribution of each instance in k-nearest neighbor instance by the sum of all K contributions to normalization. The reason for using rankings instead of actual distances is that actual distances are related to specific issues, and by using rankings, we ensure that recent instances always have the same impact on weights.

Algorithm 1 N-RRelief algorithm

```
Input: A training set $D_{train}$ consisting of a hyperparameters set $G(\theta_1, .., \theta_n)$ and performance set $C(c_1, .., c_n)$, $D_{train} = ((\theta_1, c_1), (\theta_2, c_2), ..., (\theta_n, c_n))$;
Output: Weight evaluation vector $W$ of Interaction between hyperparameters.
Initialize $N_{dc}$, $N_{dA}[\theta]$, $N_{dC\&dA}[\theta]$, $w[\theta]$, $H_{list}$ to 0.
for $i = 1$ to $m$ do
    Random selection example $R_i$;
    Selecting $k$ Neighbors $I_j$;
    for $j = 1$ to $k$ do
        $N_{dC} = N_{dC} + diff(C, R_i, I_j) \cdot d(i, j)$;
        for $\theta = 1$ to $n$ do
            $N_{dA}[\theta] = NdA[\theta] + diff(\theta, Ri, Ij) \cdot d(i, j)$;
            $N_{dC\&dA}[\theta] = NdC\&dA[\theta] + diff(C, Ri, Ij) \cdot diff(\theta, Ri, Ij) \cdot d(i, j)$;
        end for
    end for
end for
for $\theta = 1$ to $t$ do
    $W[\theta] = N_{dC\&dA}[\theta]/N_{dC} - (N_{dA}[\theta] - N_{dC\&dA}[\theta])/(m - N_{dC})$;
    According to the importance weights of hyperparameters from high to low, and taking the first t into the set of hyperparameters, calculate the interactive importance of hyperparameters in $H_{list}$:
    for $a = 1$ to $t$ do
        for $b = a + 1$ to $t$ do
            $W[\theta_m\&\theta_n] = \frac{e^{W[\theta_m]+W[\theta_n]}}{e^{\sum W[\theta]}}$;
        end for
    end for
end for
Return $W[\theta]$ and $W[\theta_m\&\theta_n]$
```

The N-RReliefF algorithm evaluates the importance measure of the interaction between hyperparameters, assigns a weight value to each hyperparameter, and evaluates how the weight is affected by the hyperparameters, so as to determine a series of the most important algorithm hyperparameters. Hyperparameters with large weight and hyperparameters combination indicate that the adjustment of these hyperparameters is very important to the performance of machine learning algorithm, while other hyperparameters with small weight mean that even if the hyperparameters are adjusted repeatedly, the influence on the performance of machine learning algorithm is not great. When the computational resources are limited, we can focus on adjusting the hyperparameters with large weights. For the hyperparameters with small weights, we can use their default values in machine learning algorithm. When sufficient computing resources are available, it is still recommended to adjust all hyperparameters. N-RReliefF algorithm can identify important hyperparameters in machine learning algorithm. The results can guide Bayesian optimization algorithm to optimize the hyperparameters and improve the performance and efficiency of the algorithm.

4 Experiment

The experiment in this section is divided into two parts. The first part is to analyze the performance of several hyperparameter optimization algorithms in machine learning hyperparameter optimization process through experiments, and select the best performance hyperparameter optimization algorithm, using its hyperparameter configuration history data to evaluate the importance of hyperparameter experiments; the second part is to use the N-RReliefF algorithm based on the hyperparameter configuration history data obtained from experiments. Evaluate the importance of super parameters and combinations of machine learning algorithms and obtain a series of super parameters which have a significant impact on the performance of the algorithms. Bayesian optimization algorithm is used to validate the effectiveness of the results obtained by N-RReliefF algorithm, which further ensures that the importance ranking of the machine learning algorithm is accurate.

4.1 Hyperparameter tuning

Experimental data and methods

In order to fully verify the superiority of Bayesian optimization algorithm, all experiments are carried out on the data set from OpenML100 [3] this section. OpenML100 is a benchmark suite that contains 100 data sets from different domains and 500 to 1000 data points with balanced distribution.

Two classification methods were analyzed on data sets from OpenML100: SVMs [5] and Random Forest [15]. For SVMs [5], two types of kernels are analyzed: radial basis function and sigmoid kernels. All algorithms use the same data pretreatment steps, including interpolation of missing data and coding of discrete features by One-Hot-Encoding. Support Vector Machine (SVM) is sensitive to the proportion of input variables, so it is necessary to standardize the input variables.

After data pretreatment, each classification method is optimized by using grid search, random search and Bayesian optimization algorithm. In order to ensure that the hyperparameter optimization method does not produce any deviation because of the configuration space of the hyperparameter, this section uses the same type and range of hyperparameter for the three hyperparameter optimization methods. The hyperparameter types and ranges of the two algorithms are shown in Tables 1 and 2.

Table 1: Hyperparameter configuration space in SVM algorithm

SVM hyperparameters	Types	Configuration space	Default value
complexity	float	[0.001, 1000.0]	[1.0]
coef0	integer	[0.0, 10.0]	[0.0]
gamma	float	$[2^{-15}, 2^{3}]$	$[2^{-15}]$
shrinking	categorical	true, false	[true]
tolerance	float	$[10^{-5}, 10^{-1}]$	$[10^{-2}]$
imputation	categorical	mean, median, mode	[mean]

Table 2: Hyperparameter configuration space in random forest algorithm

Random forest hyperparameter	Types	Configuration space	Default value
split criterion	categorical	entropy, gini	[entropy]
bootstrap	categorical	true, false	[true]
max.features	float	[0.1, 0.9]	[0.1]
min.samples leaf	integer	[1, 20]	[1]
min.samples split	integer	[2, 20]	[2]
imputation	categorical	mean, median, mode	[mean]

Experimental results

Table 3 and 4 show the SVMs performance results of different hyperparameter tuning methods under the RBF kernel function and sigmoid kernel function, respectively. Table 5 shows the performance results of random forest algorithm under different hyperparameter optimization methods. In order to increase the credibility of the results, this section uses 100 data sets from different domains to verify the average results of each index of SVMs and random forest algorithm using default parameters, grid search parameters, random search parameters and Bayesian optimization algorithm parameters. The experimental results show that Bayesian optimization algorithm can obtain the optimal performance and running time in the process of SVMs and random forest algorithm optimization.

Table 3: Average performance results of the SVM (RBF) algorithms under different hyperparameter tuning algorithms

Parameter adjustment method	Precision	$F_1 - score$	Recall	Runtime
No (using the default value)	0.22	0.30	0.47	1.8s
Grid search algorithm	0.78	0.80	0.78	40.3s
Random search algorithm	0.82	0.82	0.82	28s
Bayesian optimization algorithm	0.94	0.92	0.94	18.2s

Table 4: Average Performance Results of the SVM (sigmoid) Algorithms under Different hyperparameter Tuning Algorithms

Parameter adjustment method	Precision	$F_1 - score$	Recall	Runtime
No (using the default value)	0.54	0.37	0.49	2s
Grid search algorithm	0.80	0.82	0.81	42.9s
Random search algorithm	0.85	0.83	0.81	30s
Bayesian optimization algorithm	0.95	0.94	0.94	20.1s

Table 5: Average performance results of the random forest algorithms under different hyperparameter tuning algorithms

Parameter adjustment method	**Precision**	$F_1 - score$	**Recall**	**Runtime**
No (using the default value)	0.87	0.89	0.87	2.5s
Grid search algorithm	0.93	0.93	0.90	45.2s
Random search algorithm	0.94	0.93	0.93	35.3s
Bayesian optimization algorithm	0.98	0.97	0.97	25.5s

4.2 Importance assessment of hyperparameter

In Section 4.1, two classifiers use hyperparameter optimization N-RReliefF algorithm to generate a large number of hyperparameter configuration and performance data (including the optimal hyperparameter configuration and performance) during the operation process. These data are used to evaluate the importance of hyperparameter in the two classifiers.

Each classifier is shown by a figure and a table. Fig. 2 shows the average hyperparameter importance of hyperparameter and hyperparameter combination by bar graph. The X axis represents the hyperparameter and hyperparameter combination name, and the Y axis represents the hyperparameter importance weight. The higher the weight, the greater the impact of hyperparameters or combinations on performance. If it can not be adjusted to the appropriate value, the accuracy of the algorithm will be reduced.

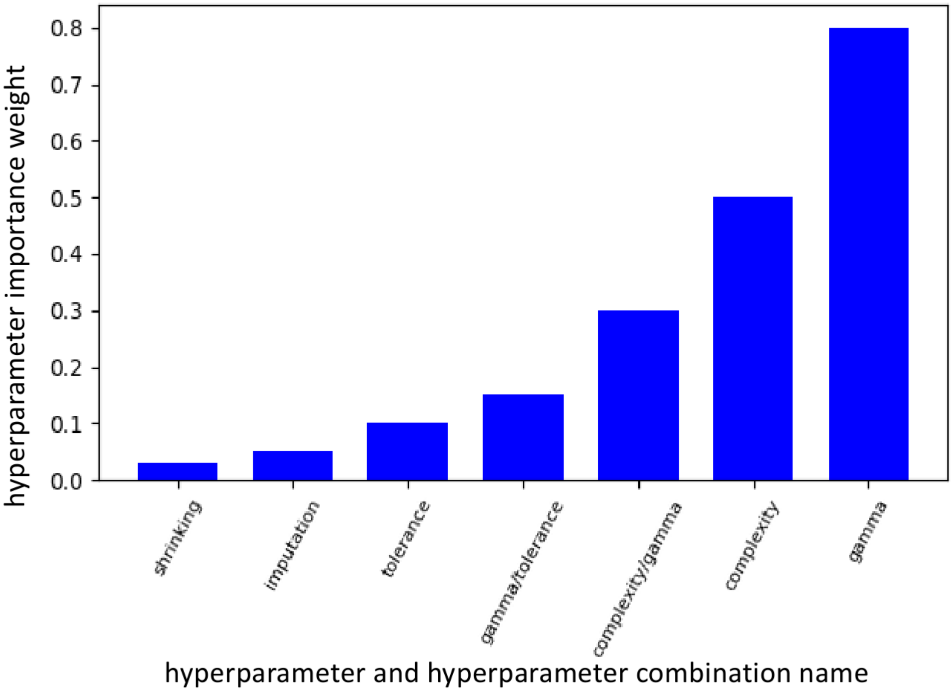


Figure 2: N-RReliefF algorithm evaluate hyperparameter and combination importance-SVM(RBF kernel)

Table 6 uses the Bayesian optimization algorithm to fix the two most important hyperparameters (using their default values) and adjust the other hyperparameters according to the most important hyperparameters selected by the N-RReliefF algorithm. The performance and running time of hyperparameter optimization are compared when all hyperparameters are adjusted and only the first three hyperparameters with great importance are adjusted according to the algorithm.

Table 6: Bayesian optimization algorithm to adjust the performance of different hyperparameters-SVM(RBF kernel)

Hyperparameters	**Precision**	$F_1 - score$	**Recall**	**Runtime**
Fixed Gamma	0.35	0.32	0.35	16.3s
Fixed Complexity	0.75	0.70	0.72	15.8s
Adjust all hyperparameters	0.94	0.92	0.94	18.2s
Adjust N-RReliefF and select top three hyperparameters	0.92	0.92	0.90	9.4s

SVM results: Fig. 2 and Fig. 3 analyze two kinds of kernel functions of SVMs, RBF kernel function and sigmoid kernel function, respectively. The experimental results clearly show that the most important hyperparameter in both cases are gamma, and the second important is complexity. This conclusion is validated by the experiments of different hyperparameter adjustment by Bayesian optimization algorithm: the hyperparameter gamma without optimizing, even if all other hyperparameters are adjusted, the classifier will get the worst performance, so it is the most important hyperparameter, complexity is the second. In addition, the performance of Bayesian optimization algorithm adjusting the first three important hyperparameters according to N-RReliefF algorithm is not different from that of adjusting all the super-parameters, and the optimization time is greatly accelerated. Fig. 3 shows that the interaction between hyperparameters Gamma and Complexity is more important than that of Complexity itself when using the sigmoid kernel. Experience shows that Gamma and Complexity are important hyperparameters in SVM (see Table.7).This paper uses a wide range of data sets to provide systematic validation for traditional experience. The least important hyperparameter for SVM's accuracy is whether to use shrinkage heuristic algorithm. The purpose of this hyperparameter is to reduce computing resources rather than improve prediction performance. According to the criteria for calculating the impact of hyperparameters on performance, its importance weight accords with the actual results.

Table 7: Bayesian optimization algorithm to adjust the performance of different hyperparameters-SVM(sigmoid)

Hyperparameters	**Precision**	$F_1 - score$	**Recall**	**Runtime**
Fixed Gamma	0.60	0.62	0.66	15.3s
Fixed Complexity	0.75	0.75	0.73	16.2s
Adjust all hyperparameters	0.95	0.94	0.94	20.1s
Adjust N-RReliefF and select top three hyperparameters	0.93	0.92	0.93	10.2s

Results of the random forest algorithm: Fig. 4 shows the experimental results of the random forest algorithm. The performance of the random forest algorithm is contributed by a small number of hyperparameters. Min. sample leaf and Max. features are the most important hyperparameters (see Table.8). In the experimental process, bootstrap is the most important hyperparameter on only a few data sets. The split criterion is the most important parameter in the data set 'scene'. Similarly, the experimental results are consistent with the Bayesian optimization algorithm validation experiment and manual parameter adjustment experience.

The final conclusion: For all classifiers, the performance changes in most cases depend

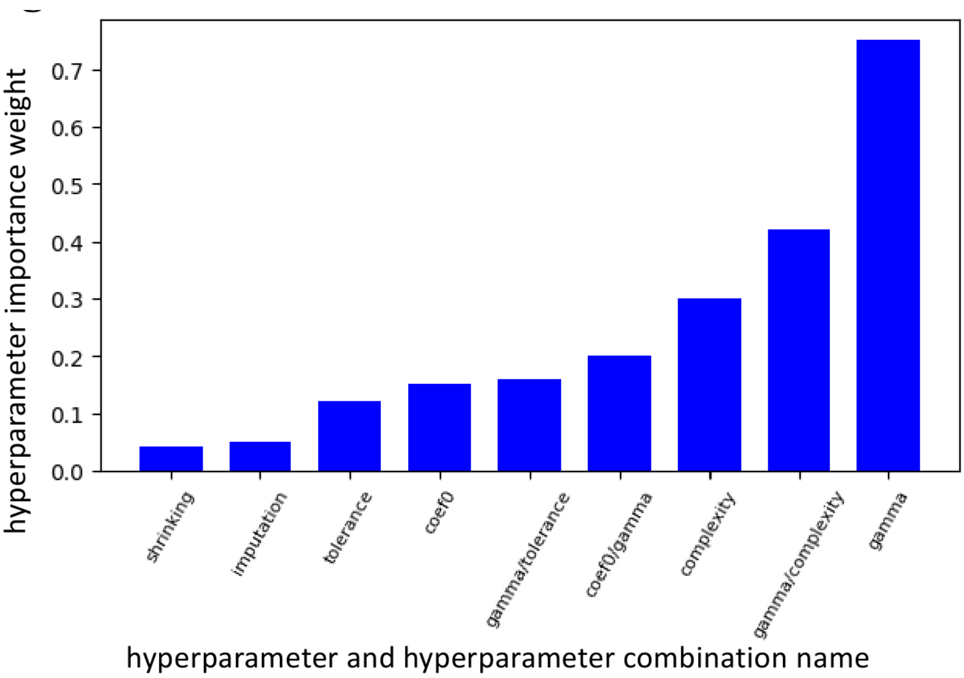


Figure 3: N-RReliefF algorithm evaluate hyperparameter and combination importance-SVM(sigmoid)

Table 8: Bayesian optimization algorithm to adjust the performance of different hyperparameters-random forest

Hyperparameters	**Precision**	$F_1 - score$	**Recall**	**Runtime**
Fixed Gamma	0.92	0.90	0.90	20.3s
Fixed Complexity	0.93	0.92	0.92	18.9s
Adjust all hyperparameters	0.98	0.97	0.97	25.5s
Adjust N-RReliefF and select top three hyperparameters	0.96	0.95	0.95	13.5s

on a small number of hyperparameters. In many cases, the same set of hyperparameters can be applied to different data sets that have the same domain and similar data characteristics. Therefore, it is necessary to understand the importance of hyperparameters in many cases, such as setting the default value of the algorithm, analyzing the automated hyperparameter optimization program and so on. In addition, understanding the importance of hyperparameters is a scientific attempt in itself, and can also provide guidance for algorithmic developers.

More interestingly, the experimental results show that the hyperparametric interpolation strategy has little effect on the performance of the classifier. In people's experience, the interpolation strategy is important, but this experiment shows that for interpolation, which strategy has little effect on the results.

It should be noted that the results provided in this section do not mean that only adjusting the most important hyperparameters and combinations is sufficient. Although Hutter [16] and others have shown that this can indeed lead to faster improvements, they also say that when there are enough computing resources, it is still recommended to adjust all hyperparameters.

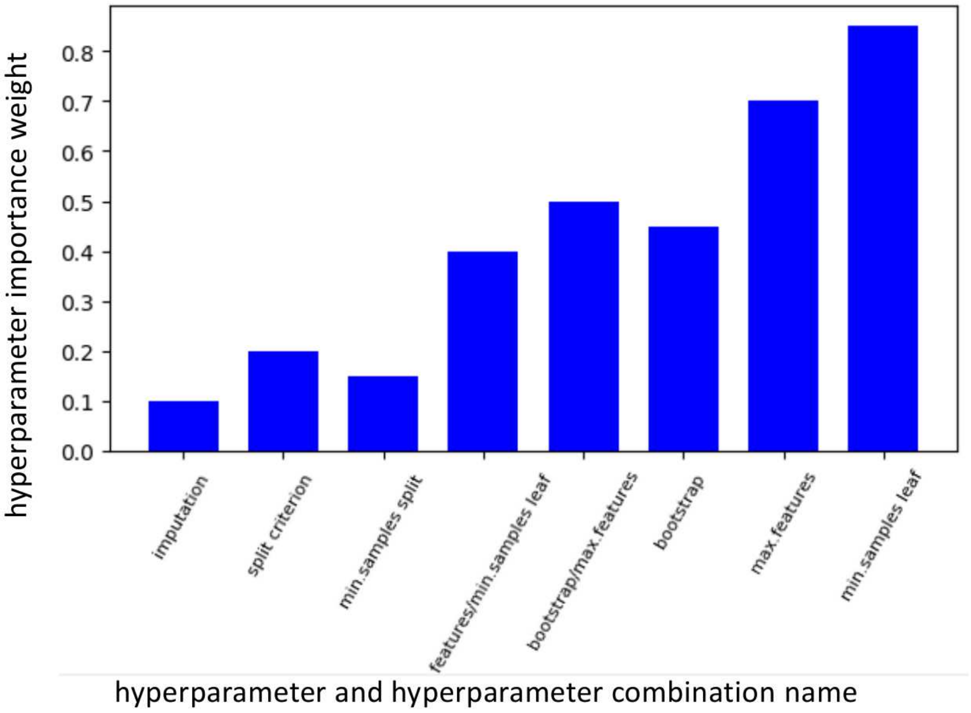


Figure 4: N-RReliefF algorithm evaluate hyperparameter and combination importance-random forest

5 Conclusion

In this paper, we compare the performance of Bayesian optimization algorithm, grid search and random search on several datasets. The results show that Bayesian optimization algorithm is superior to other two algorithms in classifier performance and running time. At the same time, N-RReliefF algorithm is used to determine the importance of the interaction between hyperparameters and hyperparameters on 100 data sets. The results show that the same hyperparameters have similar importance on different data sets. For SVMs, the hyperparameters Gamma and Complexity are the most important, and for random forest, Min. sample leaf and Max features are the most important. In order to verify the experimental results, Bayesian optimization algorithm optimizes different hyperparameters for each classifier. The results of this experiment are consistent with those of N-RReliefF, and to a large extent with popular views. A surprising result of this analysis is that data interpolation strategy has little impact on performance, which may be limited to the interpolation strategy used in this experiment. It is further proved that this conclusion needs to be studied as a whole and other interpolation strategies are added.

Future work will analyze which values of important hyperparameters are important, and provide users with the range and information of hyperparameters that can achieve better performance. In addition, it can be extended to regression and clustering algorithms to provide useful experimental support for the field of machine learning algorithm hyperparameter tuning optimization. In addition, the algorithm will be extended to the field of in-depth learning to optimize various network models (e.g. CNN, RNN models) and determine the importance of hyperparameters.

Funding

This work was funded by the Fundamental Research Funds for the Central Universities (Grant No.18CX02019A).

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Bartz-Beielstein, T. (2006). *Experimental research in evolutionary computation: The New Experimentalism*, Springer Berlin Heidelberg, 2006.

[2] Bergstra, J.; Bengio, Y. (2012). Random search for hyper-parameter optimization, *Journal of Machine Learning Research*, 13, 281–305, 2012.

[3] Bischl, B.; Casalicchio, G.; Feurer, M. et al. (2017). OpenML benchmarking suites and the openml100, ,

[4] Carlos, A.; Sellmann, M.; Tierney, K. (2009). A gender-based genetic algorithm for the automatic configuration of algorithms, *Principles and Practice of Constraint Programming - CP 2009, International Conference Proceedings*, CP 2009, Lisbon, Portugal, 142–157, 2009.

[5] Chang, C. C. C. C.; Lin, C.C.C. (2011). LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology*, 27, 1–27, 2011.

[6] Chiarandini, M.; Goegebeur, Y. (2009). Mixed Models for the analysis of optimization algorithms, *Experimental Thermal & Fluid Science*, 34(7), 972–978, 2009.

[7] Cui, J.; Yang, B. (2018). Survey on Bayesian optimization methodology and applications, *Journal of Software*, 29(10), 3068–3090, 2018.

[8] Daolio, F.; Liefooghe, A.; Sebastien, V.; Aguirre, H.; Tanaka, K. (2017). Problem Features vs. Algorithm Performance on Rugged Multi-objective Combinatorial Fitness Landscapes, *Acm Sigevolution*, 9(3), 21–21, 2017.

[9] Deng, S. (2019). Hyper-parameter optimization of CNN based on improved Bayesian optimization algorithm, *Application Research of Computers*, 2019(7), 2019.

[10] Falkner, S.; Klein, A.; Hutter, F. (2018). BOHB: Robust and Efficient Hyperparameter Optimization at Scale, *Proceedings of the35thInternational Conference on MachineLearning*, Stockholm, Sweden, 2018.

[11] Feurer, M.; Springenberg, J.T.; Hutter, F. (2015). Initializing bayesian hyperparameter optimization via meta-learning, *Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI Press*, 1128–1135, 2015.

[12] Adrian-Catalin Florea, A.-C.; Andonie, R. (2019). Weighted Random Search for Hyperparameter Optimization, *International Journal of Computers Communications* & *Control*, 14(2), 154–169, 2019.

[13] Gomes, T.A.F.; Soares, C. (2012). Combining meta-learning and search techniques to select parameters for support vector machines, *Neurocomputing*, 75(1), 3–13, 2012.

[14] Hutter, F.; Hoos, H. H.; Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration, *Learning and Intelligent Optimization -, International Conference, Lion 5, Rome, Italy, January 17-21, 2011. Selected Papers. DBLP*, 507–523, 2011.

[15] Hutter, F.; Hoos, H. H.; Leyton-Brown, K. (2013). Identifying key algorithm parameters and instance features using forward selection, *Learning and Intelligent Optimization*, 7997, 364–381, 2013.

[16] Hutter, F.; Hoos, H. H.; Leyton-Brown, K. (2014). An efficient approach for assessing hyperparameter importance, *In Proc. of ICML 2014*, 754–762, 2014.

[17] Kira, K.; Rendell, L. A. (1992). A practical approach to feature selection, *Machine Learning: Proceedings of International Conference, 1992*, 249–256, 1992.

[18] Lin, S. W.; Ying, K. C.; Chen, S. C.; Lee, Z. J. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines, *Expert Systems with Applications*, 35(4), 1817–1824, 2008.

[19] Mockus, J.; Tiesis, V.; Zilinskas, A. (1978). The application of Bayesian methods for seeking the extremum, *Towards Global Optimisation*, 2.1978, 117–129, 1978.

[20] Nannen, V.; Eiben, A. E. (2007). Relevance estimation and value calibration of evolutionary algorithm parameters, *International Joint Conference on Artifical Intelligence Morgan Kaufmann Publishers Inc*, 103–110, 2007.

[21] Probst, P.; Bischl, B.; Boulesteix, A. L. (2018). Tunability: Importance of Hyperparameters of Machine Learning Algorithms, *arXiv:1802.09596 [stat.ML]*, 2018.

[22] Reif, M.; Shafait, F.; Dengel, A. (2012). Meta-learning for evolutionary parameter optimization of classifiers, *Machine Learning*, 87(3), 357–380, 2012.

[23] Robnik-Sikonja, M.; Kononenko, I. (1997). An adaptation of Relief for attribute estimation in regression, *Fourteenth International Conference on Machine Learning, Morgan Kaufmann Publishers*, 1997.

[24] Robnik-Sikonja, M.; Kononenko, I. (2003). Theoretical and empirical analysis of relieff and rrelieff, *Machine Learning*, 53(1-2), 23–69, 2003.

[25] Snoek, J.; Larochelle, H.; Adams, R.P. (2012). Practical bayesian optimization of machine learning algorithms, *International Conference on Neural Information Processing Systems. Curran Associates Inc*, 2951–2959, 2012.

[26] Wang, J.; Zhang, L.; Chen, G. (2012). A parameter optimization method for an SVM based on improved grid search algorithm, *Applied Science and Technology*, 39(3), 28–31, 2012.

[27] Wu, J. (2017). Complex network link classification based on RReliefF feature selection algorithm, *Computer Engineering*, 43(8), 208–214, 2017.

[28] Zhang, D. (2017). High-speed Train Control System Big Data Analysis Based on Fuzzy RDF Model and Uncertain Reasoning, *International Journal of Computers, Communications & Control*, 12(4), 577–591, 2017.

[29] Zhang, D.; Jin, D.; Gong, Y.; Chen, S.; Wang, C. (2015). Research of alarm correlations based on static defect detection, *Tehnicki vjesnik*, 22(2), 311-318, 2015.

[30] Zhang, D.; Sui, J.; Gong, Y. (2017). Large scale software test data generation based on collective constraint and weighted combination method, *Tehnicki Vjesnik*, 24(4), 1041–1050, 2017.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 574-589, August 2019.

Frequent Patterns Algorithm of Biological Sequences based on Pattern Prefix-tree

L.Y. Xue, X.K. Zhang, F. Xie, S. Liu, P. Lin

Linyan Xue#, Xiaoke Zhang#, Fei Xie, Shuang Liu
College of Quality and Technical Supervision,
Hebei University,
Baoding 071002, China
lineysnow@163.com; lianlianfushi@126.com; xiefei-1214@163.com; liushuang99@hotmail.com
#Joint first authors. These authors contributed equally to this work.

Peng Lin*
College of Management,
Hebei University,
Baoding 071002, China
*Corresponding author: linpeng1982@139.com

Abstract: In the application of bioinformatics, the existing algorithms cannot be directly and efficiently implement sequence pattern mining. Two fast and efficient biological sequence pattern mining algorithms for biological single sequence and multiple sequences are proposed in this paper. The concept of the basic pattern is proposed, and on the basis of mining frequent basic patterns, the frequent pattern is excavated by constructing prefix trees for frequent basic patterns. The proposed algorithms implement rapid mining of frequent patterns of biological sequences based on pattern prefix trees. In experiment the family sequence data in the pfam protein database is used to verify the performance of the proposed algorithm. The prediction results confirm that the proposed algorithms can't only obtain the mining results with effective biological significance, but also improve the running time efficiency of the biological sequence pattern mining.

Keywords: Bioinformatics, frequent patterns, biological sequence, pattern prefix-tree, data mining.

1 Introduction

Bioinformatics is a new comprehensive cross discipline involving biology, mathematics, physics, informatics and computer science. It plays a vital role in the development of life science, and becomes the frontier of life science research. The core issue in bioinformatics is genome informatics which includes the obtaining, processing, storing, assigning and explaining of the genome information. By using computers and network as tools, based on the mathematical theory, methods and technology, genome informatics studies the biopolymers include the sequences, structures and functions of DNA and protein. The key issue in genome informatics is to understand the meaning of the order in nucleotide sequences, namely, to understand the exact locations of the genes in the chromosome and the functions of the DNA segments. These are very vital in the research of disease gene of people, the function of gene and the designing of pharmacy. To achieve those goals, pattern mining in biological data is the critical techniques [3, 7, 9].

Biological individuals have their particularity in the process of evolution, because a part of the sequence or region in the organism will affect the survival of the entire organism. Therefore, these sequence patterns will remain well-conserved in the evolutionary process. One of the vital contents of current biological sequence data analyzing is to mine proper biological sequence patterns. So far, researchers have proposed a large number of biological sequence pattern mining algorithms. These algorithms mainly find the following patterns in biological sequences:

(1) Repeat patterns in a biological sequence. For example, some continuous repeating patterns appeared in the DNA sequence called Tandem Repeats (TRS). It has been proved that these TRS play a key role in the evolution of genes. In the evolution of living individuals, using repeated sequences can help the formation of new genes [13]. Conversely, the generation of some human diseases may be caused by mutations in repeated sequences, such as DiGeorge syndrome, Williams syndrome, musculoskeletal muscular dystrophy [15], *etc.* However, we still do not fully understand these TRS and their biological functions. Therefore, finding and studying the repeat sequences in DNA sequences are very important for the establishment of repeat sequence databases and the unknown functional identification of biological sequences. In general, the above problems are included in the frequent pattern mining of biological single sequence.

(2) Conservative patterns in multiple biological sequence sets. Some sequence regions in the process of genetic variation in organisms will affect the survival of organisms, so these sequences will be highly conserved during the entire evolution of organisms. For example, most or even all of the sequences from the same family sequence set (such as the protein family) often contain conserved sequence pattern regions that play a vital role in the structure and function of protein [2]. Important functional sequences (transcription factor binding sites), which would be generally located in upstream region of co-expressed gene sequence, tend to be more conserved and can regulate the expression of genes [17]. These examples can all be seen as mining their conservative patterns in multiple biological sequences.

(3) Sequence patterns with different frequencies appear in sequences in multiple biological sequence sets. A repeating sequence is called a copy. In the human population, there are often differences in the number of repeats (the number of copies) of the repeat sequence, known as repeat copy number polymorphisms. This feature is widely used in genetic diversity analysis, individual identification, genetic diagnosis, genetic mapping and other applied research. For example, Saghai et al. conducted an experiment to verify the polymorphism through the RFLP linkage map [16]. In the experiment, they compared and analyzed the conserved regions covering 24% of rice gene sequences and 17 conserved regions covering 31% of barley gene sequences. They found that 72% of single copy genes in barley were similarly expressed in rice in a single copy. Similarly, about 60% of rice single-copy sequences are found in barley. This result shows that the difference between the grass crops is caused by the difference in repeat sequences, not due to structural differences in the genes. Excavating the repetitive patterns in the multi-sequences of organisms and their frequency of occurrence in each sequence has guiding significance in genetic recognition and so on [18]. This type of problem can be incorporated into the study of frequent patterns mining of biological multi-sequences.

2 Related works

At present, there are two main kinds of computational methods for studying biological sequence pattern discovery. Each kind of algorithm has a different search strategy. One kind of algorithm uses a heuristic search strategy and it is an approximate algorithm. In the artificial intelligence field, most machine learning methods use heuristic algorithms. In sequence pattern mining, such methods mainly include Gibbs sampling algorithm, EM method [11], MEME algorithm [4] and so on. This kind of algorithm is usually an iterative process, and it gets better solutions through iterations. In the algorithm, some approximate description of the sequence pattern information is needed to determine the quality of a certain measurement standard. In the iteration, the solution is optimized according to the criterion, and it is judged whether the iterative termination condition is satisfied. The advantage of this kind of algorithm is that the computational complexity is low and it is suitable for searching for a longer sequence pattern. However, the disadvantage is that the solution it obtains may be a local optimal solution, and it may not be able to obtain a global optimal solution. However, a large number of application practices have proved that such approximate solutions obtained by machine learning algorithms can be used to solve practical application problems.

Since the frequent pattern mining was defined by Agrawal and Srikant in 1995 [5], related research has become an important field of data mining and has received extensive attention from researchers. They proposed many algorithms, some of which are suitable for efficient mining of large-scale sequential patterns, such as SPADE, FreeSpan, Prefixspan, GSP, Apriori and so on. However, biological sequences are different from sequence data such as transaction sequences. When applying the above algorithm directly to biological sequence data, there are many difficulties and problems to be solved. For example, the meaning of fuzzy matching between patterns is difficult to understand, the mining results cannot meet the needs of biological research, some pruning strategies or data structures cannot be effectively applied to biological sequence data, and algorithm for data volume scalability. It is necessary for biological sequences to design frequent pattern detection algorithms. The existing algorithms mainly have the following categories:

(1) Tandem Repeats mining algorithm. Tandem repeats are a special type of sequence pattern, which is a subsequence that is arranged end-to-end in a DNA sequence, has repeating units in a string, and has a frequency exceeding a certain threshold. Tandem Repeats include Perfect Tandem Repeats (PTR), Longest Pattern Repeats (LPR), Approximate Tandem Repeats (ATR) and so on. Jiang et al. in [8] proposed the definition of the LPR for the exact search for new repeats, and designed an LPR search algorithm based on the subsequent array structure. KurtZ et al. [10] gave REPuter algorithm that depended on the data structure of the suffix tree. The repeated sequences are mined by pairwise alignment of subsequences. However, these algorithms are still difficult to find for frequently occurring repeats in DNA sequences [1, 6]. In addition, Won et al. proposed a mosaic silhouette algorithm to identify repeated sequences [19].

(2) Mining sequence models with conditional restrictions. In practical applications, we often want to mine sequence models with conditional constraints based on the characteristics of biological sequences and the needs of application problems. Such constraints usually include the need to mine patterns of biological sequences that may contain intervals of any length, to allow fuzzy matches between sequence patterns, and so on. The existing sequential pattern mining algorithm with some constraints does not consider the biological sequence's various constraint features. Liao et al. proposed a two-stage algorithm that can mine sequence patterns containing arbitrary length intervals [12]. The algorithm is divided into two phases: the first phase searches for all short frequent patterns, and these is no gap in these patterns; the second phase generates long patterns containing these short frequent patterns. The algorithm can use the short frequent mode information to reduce the search time of the spaced global mode. Although the algorithm can mine more sequence patterns that are more biologically meaningful, it takes more time to generate long patterns.

(3) Frequent subtree mining. With the continuous expansion of data mining applications, frequent patterns of mining objects are constantly diversified. There are sequences, transaction itemsets, and complex data structures such as graphs and trees to adapt to web mining, biological structure data mining semi-structured document mining. Therefore frequent subgraph mining and frequent subtree mining have become important research fields in frequent pattern mining. After several years of research, there have been some methods for mining frequent patterns such as trees and graphs. In the research of frequent subtree mining algorithms, the identification of tree isomorphism and the matching of tree patterns are the two key issues to be solved. In the frequent subtree mining algorithm, it is determined whether the tree in the database has a subtree that is isomorphic to a certain known tree. Let the tree in the database be T and the known tree be P, $/T/\geq/P/$. If there is a one-to-one correspondence between the vertices of P to T, and the edges of the corresponding vertices have the same relationship, T is a subtree of P.

In the tree pattern matching problem of biological data, the trees T and P are ordered trees, and each vertex and edge have markers. At this point, the isomorphism of the subtree requires that the corresponding vertices have the same mark [14].

3 Single sequence frequent pattern mining algorithm

3.1 Basic mode

First of all, it is stipulated that DNA sequences always consist of four characters *A, C, G, T* and *\$* as the terminator. On this basis, the relevant definitions of patterns are given.

Definition 1. For alphabet $\Sigma = \{A, C, G, T\}$, pattern $P =< p_1, p_2, \ldots, p_n >$ and pattern $P' =< p'_1, p'_2, \ldots, p'_n >$ are given, approximation degree is defined as follows:

$$\text{Approximation\_degree}\left(P, P'\right) = \frac{|E|}{\text{length}(P)}$$

where $E = \{j | p'_j = p_j, j = 1, 2, \ldots, n\}$, it represents the set of same characters between two patterns. The approximation degree takes into account not only the Hamming distance between patterns, but also the influence of pattern length. Assuming that the Hamming distances between patterns are the same, the pattern lengths are 2 and 10, respectively, it is obvious that if the pattern length is 10, it means that the two patterns are more similar. When the degree approximation is 1, it means that the two patterns match perfectly.

Definition 2. For alphabet $\Sigma = \{A, C, G, T\}$, pattern $P =< p_1, p_2, \ldots, p_n >$ and pattern $P' =< p'_1, p'_2, \ldots, p'_n >$ are given, Approximation_match(P, P') represents whether pattern P and pattern P' satisfy the minimum approximation specified by the user.

Definition 3. (Frequent approximation patterns). Given sequence S, pattern $P =< p_1, p_2, \ldots, p_n >$ is a frequent approximation pattern in sequence S if and only if $\sum_{E=1}$ Approximation_match$(P, P') \geq m$, where

(1) P_i is a substring of S,

(2) $|P_i| = |P|$,

(3) m represents the minimum frequent threshold specified by the user,

(4) For $P_i = S[a \ldots b]$, $P_j = S[c \ldots d]$, where $i \neq j$, if Approximation_match(P, P_i)=1 and Approximation_match$(P, P_i) = 1$, then it must be $b < c$ or $d < a$.

Definition 4. (Support of frequent approximation patterns). Given sequence S, pattern $P =< p_1, p_2, \ldots, p_n >$ is a frequent approximation pattern in sequence S, then we call

$$\text{support}(P) = \frac{\sum_{i=1}^{n} \text{ Approximation\_match}\left(P, P_i\right)}{[\text{length}(S) / \text{ length}(P)]}$$

as support of frequent approximation patterns P on sequence S. The higher the support degree, the higher the frequency of pattern occurrence in the sequence. When the support degree is 1, the pattern is a frequent and accurate pattern in the sequence.

Lemma 5. *0≤Approximation_degree(P, P′)≤1*

It is obvious that Approximation_degree(P, P′)≥0, according to Definition 1, for pattern $P =< p_1, p_2, \ldots, p_n >$ *and pattern* $P' =< p'_1, p'_2, \ldots, p'_n >$, $I = \{i | p'_i = p_i, i = 1, 2, \ldots, n\}$, *it is obvious that* $|I| \leq \text{length}(P)$, *so Approximation_degree(P, P′)≤1.*

Lemma 6. *$0 \leq sup(P) \leq 1$*

It is obvious that $sup(P) \geq 0$, according to definition of frequent approximation patterns, for $P_i = S[a \dots b]$, $P_j = S[c \dots d]$, if Approximation_match(P, P_i)=1, and Approximation_match (P, P_j)=1, then it must be $b < c$ or $d < a$, so $sup(P) \leq 1$.

P is a frequent approximation pattern in sequence S, if there are m patterns P_1, P_2, $\dots$, P_m in sequence S, where $P_1 = S\left[a_1, a_1'\right]$, $P_2 = S\left[a_2, a_2'\right]$, $\cdots$, $P_m = S\left[a_m, a_m'\right]$ $(a_1 < a_2 < \cdots < a_m)$, they are all satisfied Approximation_match(P, P_i)=1, in order to maximize the number of non-overlapping patterns in S sequence, the first non-overlapping pattern must be P_1.

According to definition of frequent approximation patterns, $|P_i| = |P|$. let $|P| = l$, then $P_1 = S[a_1, a_1+l-1]$, $P_2 = S[a_2, a_2+l-1]$, $\cdots$, $P_m = S[a_m, a_m+l-1]$. If the first non-overlapping pattern is not $S[a_1, a_1 + l - 1]$, it is the i-th $S[a_i, a_i + l - 1]$, where $a_i > a_1$. If $a_{i+1} - a_i \geq l$, then the second non-overlapping pattern is $S[a_i + 1, a_{i+1} + l - 1]$. If $a_{i+1} - a_i < l$, then a_{i+2} and a_i are compared, if $a_{i+1} - a_i < l$, then a_{i+3} and a_i are compared, until $a_{k+1} - a_i \geq l$. So the second non-overlapping pattern is $S[a_k, a_k + l - 1]$. And so on, the n-th non-overlapping pattern is obtained.

Obviously, the second non-overlapping pattern will not overlap with the first non-overlapping pattern. Because $a_i > a_1$, it doesn't overlap with pattern $S[a_1, a_1 + l - 1]$. The method can be used to clip candidate patterns and quickly find frequent approximate patterns satisfying conditions.

3.2 Basic mode table

After getting all the basic patterns of a sequence S, a basic pattern table of S can be constructed. For ease of searching, it needs to sort all the basic patterns of S by the lexicographic order of the characters in $|\Sigma|$.

For example, for basic pattern of S="yxzxxyzyxy", after sorting, it can get the basic schema list of S, it is shown in Table 1. In Table 1, each item is in the form of (Num, S_m, loc), where: Num represents the number of item; S_m represents the basic pattern; loc is starting position of the basic pattern S_m in S.

Table 1: Basic mode table for S

Num	S_m	**loc**
1	x	3
2	xy	7
3	xyzy	6
4	xz	3
5	y	9
6	yx	7
7	yxzxx	2
8	yz	4
9	zxxy	1
10	zyxy	6

Of course, there may be duplicates in the basic pattern, such as: S'="xxyxxyxyxy", the basic pattern "x","yx" all appear twice, "x" appears 4 times. In this case, the above basic model table can be improved and designed as shown in Table 2.

In Table 2, each item is still in the form of (Num, S_m, input, loc), but where loc is the set of starting positions of the basic pattern S_m in S.

Table 2: Basic mode table for S'

Num	S_m	**loc**
1	x	1 3
2	xy	1 3 6 8
3	y	9
4	yx	5 7
5	yxx	3

3.3 Construct basic frequent pattern prefix tree

Definition 7. For a basic pattern table, the basic pattern prefix tree which the basic pattern table corresponds to is a rooted tree, whose path from root to each leaf node represents a basic pattern. Every side of path represents a substring, which represented by edges of the same node do not have the same prefix. A basic pattern is achieved through sequentially arranging substrings represented by edges on the path in the basic pattern prefix tree.

For the basic model Table 1, the basic pattern prefix tree shown as Figure 1 could be constructed by using the above recursive algorithm.

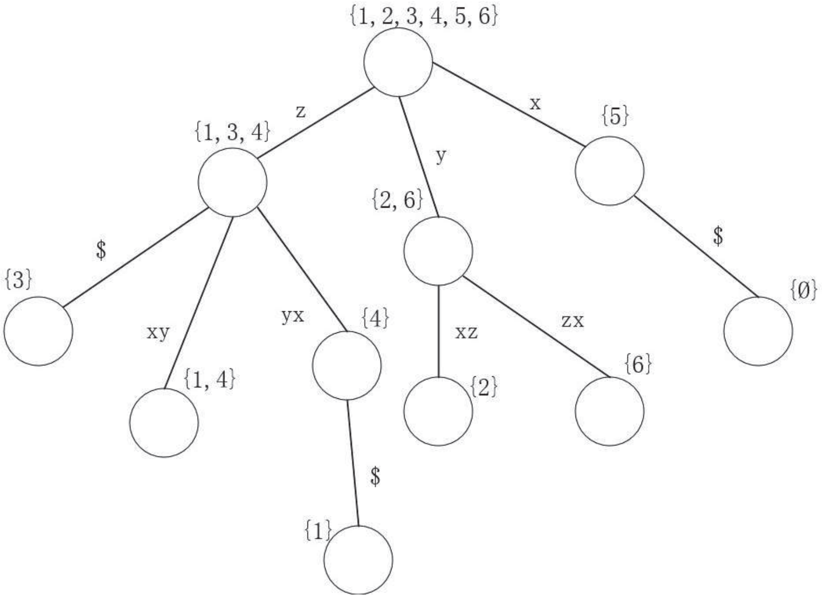


Figure 1: The basic pattern prefix tree corresponding to S

In the tree, each node has a set of starting points $\{l_1, l_2, \cdots, l_k\}$. Its meaning is: x is the substring represented by the edge of the node connected to its father node, then $\{l_1, l_2, \cdots, l_k\}$ represents the substring x in the initial position collection of S. For root node, $\{l_1, l_2, \cdots, l_k\}$ represents the set of initial position of the empty string in S, that is, the collection $\{1, 2, \cdots, |S|\}$ for all positions in S.

By the above analysis and its definition, a novel recursive algorithm to construct a basic pattern prefix tree is designed. Starting from root node, the prefix subtree is constructed layer by layer for each vertex. The core is to apply the following node-extend(S, d) extension algorithm to a node. Since we need to calculate displacements for the set of starting positions in this algorithm, we define the addition of sets and numbers for this purpose.

Definition 8. Given the set $t = (t_1, t_2, \cdots, t_k)$ and the number l, their addition result is a set: $t + l = \{t_1 + l, t_2 + l, \cdots, t_k + l\}$.

The node's extension algorithm node-extend(S, d) establishes a prefix tree with d as the root for collection S of basic pattern string sets. The basic pattern table of the string H is $S(H)$, then constructing the basic pattern prefix tree of H is the following recursive called Node-extend$(S(H), root)$.

3.4 Pruning of the basic pattern prefix tree

Using the basic pattern prefix tree, basic frequent patterns can be mined. Firstly, we trim the basic pattern prefix tree as follows. If loc=$\{l_1, l_2, \cdots, l_k\}$ and loc is less than minloc_sup in a pattern (Num, S_m, loc), the node and the subtree rooted from the node can be subtracted. Because in a pattern, if its subpattern is not frequent, then according to the nature of Apriori algorithm, the pattern itself must not be frequent. If a basic pattern is not frequent, all patterns including it would be certainly not frequent. Deleting infrequent nodes and their subtrees in the basic pattern prefix tree will delete the infrequent subpatterns.

Given loc_sup is 2, after pruning the basic pattern prefix tree in Figure 1, the prefix tree which is shown as Figure 2 can be obtained.

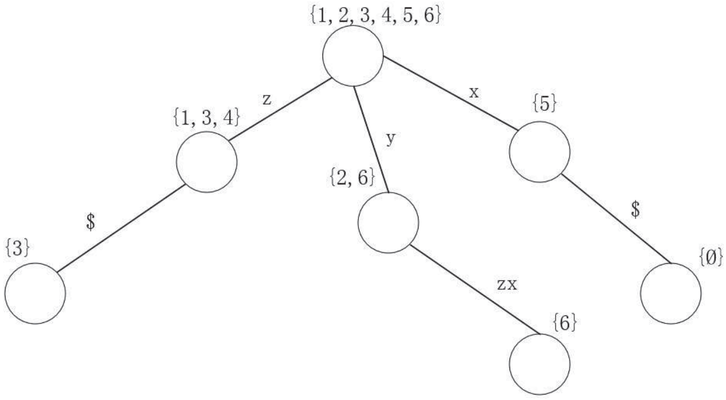


Figure 2: Prefix tree after pruning

From the prefix tree we can know that the original string. The frequent patterns in S="aabaababab" are: the frequency of "a" is four, the frequency of "b" is four, the frequency of "c" is two, the frequency of "ab" is two, and the frequency of "ba" is two.

3.5 General frequent pattern mining

The basic frequent mode is limited to that there is no letter in the mode that is the same as the start letter. The above basic pattern prefix tree can only mine the basic frequent mode, and cannot mine the general frequent mode. In order to enable it to mine general frequent patterns, we designed a general frequent pattern detection algorithm based on basic sub-pattern prefix tree. The algorithm is a recursive algorithm. From root node, every node is model-expanded layer-by-layer. The kernel of the algorithm will be to apply the expansion algorithm Span(s, E_s, a) to a node. There is a pattern s, whose set of end positions in the string S is Es. In the basic pattern prefix tree, the last node of the path of the pattern s is a. The algorithm could detect all the frequent patterns prefixed by s.

In order to detect all frequent patterns in the string S, Span$(\Phi, E, root)$ can be called where $E = \{1, 2, \cdots, |S|\}$.In this way, the algorithm starts with an empty string, expands from tree to tree, and records the extended set of end positions at each layer. When extended to a leaf node, it will return to the root node and continue to expand until the extended substring is infrequent.

By synthesizing the above several steps, the single sequence pattern mining for biological data algorithm for single biological sequence mining can be obtained. The algorithm quickly detects all basic frequent patterns firstly, and then implements pattern growth on this basis to obtain all frequent patterns of the sequence.

The flow of the algorithm is defined as:

Algorithm 1

Step 1. A candidate pattern P is extracted from set C in turn, if pattern P is already in the approximate frequent pattern set S, the next candidate pattern is selected. Otherwise, the candidate pattern P is compared with other element C $[index]$ of set C.

Step 2. If Approximation_degree($P, C[index]$) is greater than minimum approximation, then the number of repetitions of candidate pattern P is increased by 1, $index = index + i$. candidate pattern P continues to compare with element C $[index]$ of set C. Otherwise $index = index + 1$, candidate pattern P continues to compare with element C $[index]$ of set C.

Step 3. Repeat step 2 until index is greater than or equal to the maximum subscript in set C, repetition times and support degree of candidate pattern P are counted, if the repetition times and support degree of candidate pattern P satisfy the specified minimum frequent threshold m and the minimum support degree minsupport, then candidate pattern P is a frequent approximation pattern and is added to the approximate frequent pattern set S.

Step 4. Repeat step 1 until every element in set C is taken out.

4 Multiple sequence frequent pattern mining algorithm

Above algorithm for mining single sequence frequent patterns can be extended for mining multiple sequence frequent patterns.

Definition 9. Distribution Support was set a collection of biological sequences D and subsequences T, the number of sequences containing subsequences T in D is called the distribution support of T. The distribution support degree of sub-sequence T is the quantity of sequences containing subsequence T in D, denoted as dis_sup$D(T)$.

Definition 10. Given biological sequence set D and subsequence T, if the sub-sequence support distribution dis_sup$D(T)$ is greater than or equal to mindis_sup, T is said to be a frequent pattern.

For understanding the process of constructing a multiple sequence basic model table and its prefix tree easier, we would give the following definition:

Definition 11. (Set vector) There are vectors $T = (T_1, T_2, \cdots, T_n)$ where each T_i is a set, and T is a set vector.

Definition 12. (Support function of the set vector) For the set vector T, its support function $F(T)$ is defined as the number of non-empty sets in T,

$$\mathrm{F(T)} = \left|\{\mathrm{T_i}|\mathrm{T_i} \neq \emptyset; 1 \leq \mathrm{i} \leq \mathrm{n}\}\right|. \tag{1}$$

Definition 13. (Operation of set vector) If there are set vectors $S = (S_1, S_2, \cdots, S_n)$ and $T = (T_1, T_2, \cdots, T_n)$, the intersection of the two vectors is defined as:

$$\mathrm{T} \cap \mathrm{S} = (\mathrm{T_1} \cap \mathrm{S_1}, \mathrm{T_2} \cap \mathrm{S_2}, \ldots, \mathrm{T_n} \cap \mathrm{S_n}). \tag{2}$$

Definition 14. (Addition of Set Vectors and Numbers) With the set vector $T = (T_1, T_2, \cdots, T_n)$ and the number l, there is

$$T + l = T_1 + l, T_2 + l, \cdots, T_n + l. \tag{3}$$

"+" represents the addition of sets and numbers as defined in Definition 3.

4.1 Multiple sequence basic pattern table

For the mining of multiple sequence frequent patterns, firstly we would intercept all the basic patterns in each sequence, and then sort the basic pattern tables of each sequence to obtain the basic pattern table of multiple sequences in the sequence set.

Suppose there are four sequences in the sequence set D. S_1="$xyzyxz$", S_2="$xzyzx$", S_3="$yzyxyz$", S_4="$xzyxyz$". Algorithms 1 is invoked for each sequence. The basic mode table for each sequence of D is shown as Table 3.

Table 3: Basic patterns of multiple sequences

Basic pattern table of S_1

Num	S_m	**loc**
1	xyzy	1
2	xz	4
3	yxz	3
4	yz	3
5	z	5
6	zyx	6

Basic pattern table of S_2

Num	S_m	**loc**
1	xy	4
2	xzyz	1
3	y	5
4	yzx	2
5	zxy	3
6	zy	2

Basic pattern table of S_3

Num	S_m	**loc**
1	xyz	3
2	yx	4
3	yz	1 4
4	z	5
5	zyxy	2

Basic pattern table of S_4

Num	S_m	**loc**
1	xyz	3
2	xzy	1
3	yx	2
4	yz	4
5	z	5
6	zyxy	3

In order to mine multiple sequence frequent patterns, after we get the basic pattern tables of multiple sequences, it must integrate the tables to get a merged multiple sequence pattern table. For example, the basic pattern table of 4 sequences in Table 3 is integrated to obtain basic frequent pattern table of multiple sequence sets similar to Table 1, it is shown in Table 4.

Table 4: Basic pattern table after merging

Num	S_m	**loc_set**
1	x	{1,5},{1,5},{4},{4}
2	xy	{1},{5},{4},{4}
3	xyz	{1},∅,{3},{3}
4	xz	{4},{1},∅,{2}
5	xzy	∅,{1},∅,{1}
6	y	{1,3},{2,5},{1,2,4},{2,4}
7	yx	{3},∅,{2},{3}
8	yz	{2},{2},{4},{4}
9	z	{2,5},{1,3},{1,4},{2,4}
10	zx	{2},{1},{2},{3}
11	zyx	{2},∅,{1},{1}
12	zyxy	∅,∅,{2},{1}

From Table 4 basic frequent patterns could be easily discovered. Let (Num,S_m,loc_set) be the term of a basic mode S_m, then the frequency of S_m is the support function F(loc_set) of the set vector loc-set. For example, if dis_sup is set to 2, then according to Table 4 we can easily tap out the basic frequent patterns: the frequency of "x" is four, the frequency of "xy" is four, the frequency of "xyz" is three, the frequency of "xz" is three, the frequency of "xzy" is two; the frequency of "y" is four, the frequency of "yx" is three, the frequency of "yz" is four; the frequency of "z" is four, the frequency of "zy" is four, the frequency of "zyx" is three, the frequency of "zyxy" is one.

4.2 Multiple sequence basic frequent pattern prefix tree

Assuming that the basic pattern table of the multiple sequence set D is $D(H)$, then constructing the basic pattern prefix tree of D can invoke the node's extension algorithm. The algorithm is a recursive algorithm that can be invoked with node-extend $(D(H), root)$. In contrast to the mining of single sequence frequent patterns, the "+" in the "(P',loc+$|x_i|$)" of the algorithm represents the set vector and number defined in the Definition of 12.

Since the basic patterns in Table 4 are all frequent patterns, the following recursive algorithm is used to construct the following multiple sequence basic frequent pattern prefix tree.

Assume that there are k sequences $1, 2, \cdots, k$ in D. In the tree, each side of x represents a character or substring; x is connected to the node in the direction of the leaf is b, there is a set vector $T^{(b)} = (T_1^{(b)}, T_2^{(b)}, \ldots, T_k^{(b)})$ on b node represents the mode prefixed with x in each sequence in D. $T_i^b = \{l_1, l_2, \ldots, l_{ki}\}$ is the collection of initial positions of x in sequence S_i. A basic frequent pattern in Table 4 can be obtained by sequentially arranging the characters represented by the nodes on the path from root to every leaf node.

By synthesizing above several steps, multiple sequence pattern detection for biological data algorithm for multiple biological sequence frequent pattern mining can be obtained. The algorithm quickly detects all basic frequent patterns firstly, then implements pattern growth on this basis to obtain all frequent patterns of sequence set. Algorithm's framework is as follows:

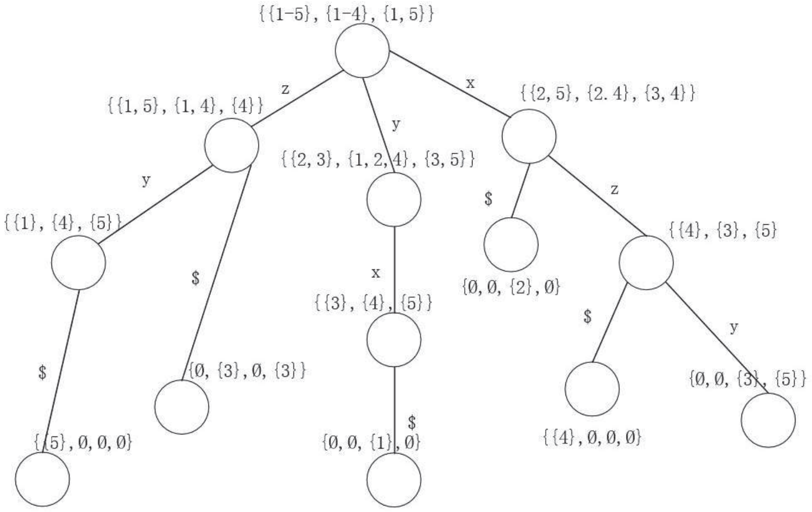


Figure 3: The basic frequent pattern prefix tree corresponding to D

Algorithm 2

```
Multiple sequence pattern mining for biological data algorithm
Input: Biological sequence $S$, minimum local support mindis_sup;
Output: All frequent pattern sets Freq_set;
Begin
1. For each sequence $S_i$ in $S$ do
2. Intercept($S$); /* The intercept algorithm is used to get all the basic pattern string sets $H$ of sequence $S$.*/
3. Sort($S_{mi}, S''_{mi}$); /* Using the algorithm, the basic schema table $S_i(H)$ in order of lexicographic order can be gotten.*/
4. End For
5. Merge the basic schema tables of multiple sequences and sort them.
6. The basic pattern frequency table of multiple sequence sets is constructed and all basic frequent pattern sets $H$ are extracted;
7. Node-extend ($S(H), root$) is recursively called to construct the basic frequent pattern prefix tree $T$;
8. MFreq-Mining($S, root$);
End.
```

5 Experimental results and analysis

In order to verify effectiveness of the algorithm, two sets of experiments were compared. First set of experiments compares the traditional algorithm with the proposed two fast and efficient biological sequence pattern detection algorithms. It is mainly used for verifying that these algorithms change with the supportability threshold, which has little impact. The second set of experiments compares the traditional algorithm with the proposed algorithms to verify that the proposed algorithms in this paper have better elapsed time efficiency under the same supportability threshold.

Under the same biological sequence set, with the increasing of support threshold, running time changes of Prior, BioPM, and the single sequence frequent pattern mining algorithm are shown in Figure 4.

From Figure 4, it can be seen that the elapsed time of each of the three algorithms will gradually decrease with increasing of the support threshold. However, overall elapsed time of the single sequence frequent pattern mining algorithm changes steadily, and it is obviously smaller than the other two algorithms. Especially when the threshold is small, the Apriori algorithm will generate a large number of candidate modes and interference modes during the mining process, which inevitably leads to a high complexity of the space-time of the algorithm. Similarly, the BioPM algorithm needs to construct a projection database frequently during the mining process, and there are also a large number of short-term generations, which greatly affects the efficiency of algorithm. The single sequence frequent pattern detection algorithm starts from length of basic pattern. It prunes basic pattern prefix tree in the process, avoiding the generation of a great quantity of short biological patterns and candidate patterns to speed up the operation speed and improve the efficiency.

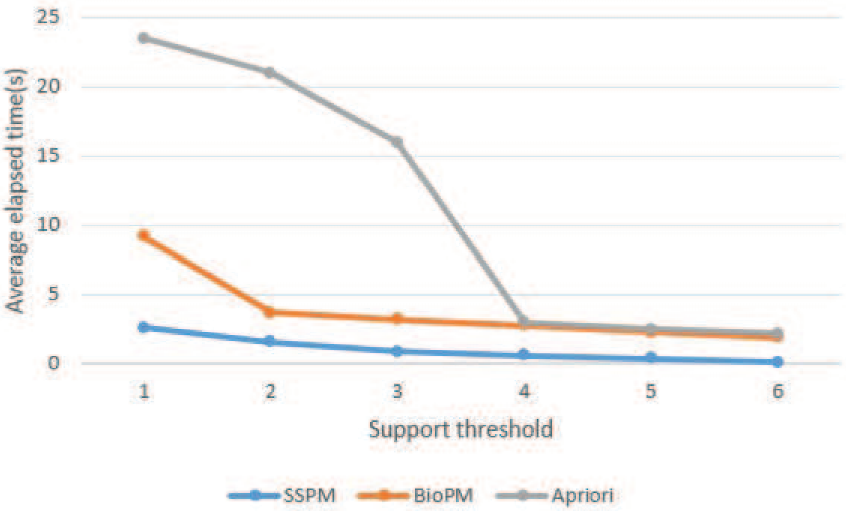


Figure 4: The relation between support threshold and average elapsed time

The following experiment is used to verify that the single sequence frequent pattern mining algorithm has better excavation efficiency under the same support degree threshold. The experimental data were from 10 families in the Pfam protein database. The same or similar part of the length is selected as the test set, and set a 15% support threshold for them. From the 100-sequences, the sequence was gradually increased. The patterns were mined using the BioPM and the single sequence frequent pattern mining algorithm respectively, and the overall time of pattern mining of all sequences was obtained. As the number of sequences increases, the overall elapsed time trends of the two algorithms are shown in Figure 5.

From Figure 5, it can be seen that under certain fixed support threshold conditions, the running time of both the BioPM algorithm and single sequence frequent pattern mining algorithm will increase as the quantity of sequences increases. However, running time of the single sequence frequent pattern mining algorithm is much smaller than that of the BioPM algorithm, and the trend of change is always stable. The reason is that the BioPM algorithm needs to construct a projection database frequently during the mining process and also a large number of short-term generations causes the complexity of the algorithm's space-time and increase the efficiency. However, the single sequence frequent pattern mining algorithm is to start frequent pattern mining from the basic pattern length, which avoids the elapsed time overhead of generating a great quantity of short patterns.

Because classic Apriori algorithm, BioPM algorithm, and multiple sequence frequent pattern

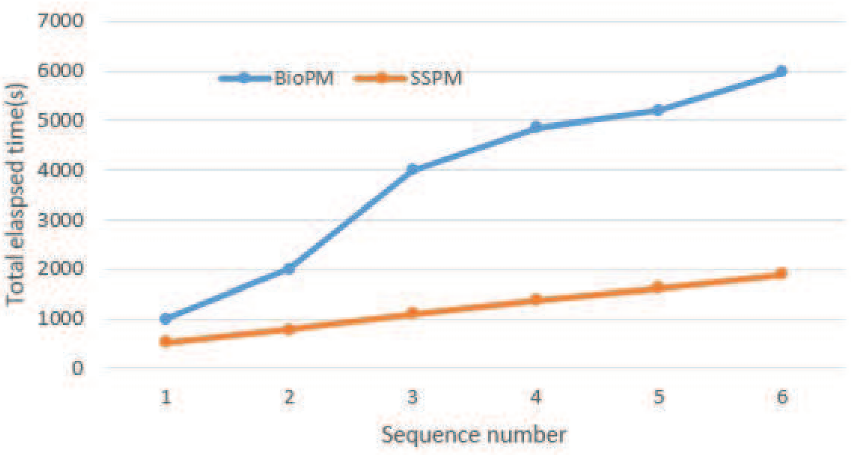


Figure 5: The relationship between the number of sequences and the time

detection algorithm are deterministic algorithm of multiple sequence frequent patterns, the set of frequent patterns mined for the same biological sequence collection and support threshold is exactly the same. In order to verify the performance of the multiple sequence frequent pattern mining algorithm, two sets of experiments were conducted and then compared based on the experimental results. In the first set of experiments, these algorithms are compared with the multiple sequence frequent pattern mining algorithm to verify that the multiple sequence frequent pattern mining algorithm is less affected by changes in the support threshold. The second set of experiments compares the BioPM algorithm, the MBioPM algorithm, and the multiple sequence frequent pattern mining algorithm. It is verified that under the same premise (with the same support threshold), the elapsed time efficiency of the multiple sequence frequent pattern mining algorithm is better, and it has superior excavation performance.

Through the group of experiments, it can be confirmed that the multiple sequence frequent pattern mining algorithm is affected less by the change of the support threshold set by the user. Experimental data were sampled from 3 families in the Pfam protein sequence database (G-alpha, Calici Coat, Glyco_hydro_19). A subset of the same or similar lengths (out of a total of 50) are selected as test sets to ensure that this algorithm is suitable for different types of sequence sets. Under each of the specific support conditions, 50 sequences of data were tested using four algorithms respectively. Under the same set of biological sequences, as the support threshold is increased, the elapsed time changes of the Prior, BioPM, MBioPM, and the multiple sequence frequent pattern mining algorithm are shown in Figure 6.

From Figure 6, it can be seen that the elapsed time of the four algorithms will gradually decrease as support threshold increases. However, the trend of overall elapsed time of the multiple sequence frequent pattern mining algorithm is relative stability, especially when the threshold becomes small, the elapsed time will be obviously less than the other three algorithms. The main reason for this result is that the Apriori algorithm will generate a great quantity of candidate modes and interference patterns during mining process, which will inevitably lead to a higher complexity of the space-time algorithm; the BioPM algorithm requires frequent construction of the projection database during the mining process, and there are also a large number of short-mode generations, which greatly affect the efficiency of the algorithm; the multiple sequence frequent pattern detction algorithm is started from length of basic model to avoid generating a great quantity of short biological models, through the merge operation can quickly basic frequent pattern detection and use basic frequent pattern prefix tree for pattern growth to improve

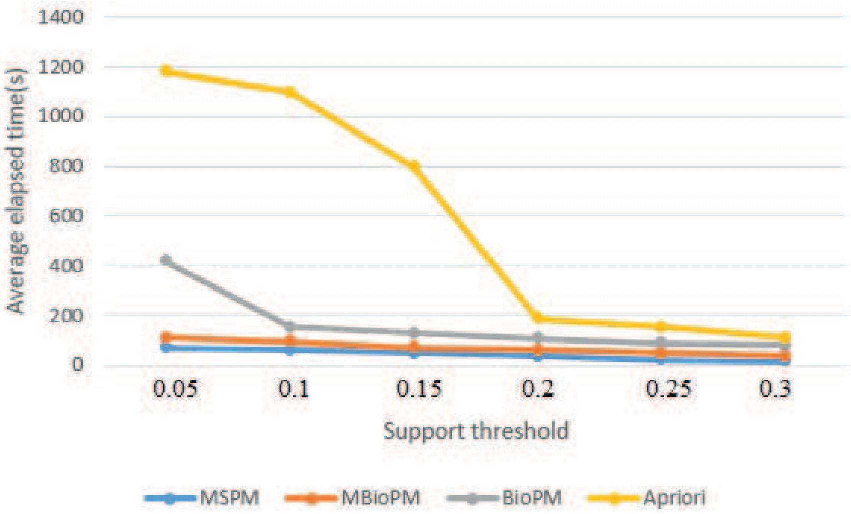


Figure 6: The relation between support threshold and average elapsed time

efficiency.

The set of experiments is used to verify that the multiple sequence frequent pattern mining algorithm has better excavation efficiency under the same support degree threshold. The experimental data were from 10 families in the Pfam protein database. The same or similar part of the length is selected as the test set, and set a 15% support threshold for them. From the 100-sequences, the sequence was gradually increased. The patterns were mined using the BioPM and the multiple sequence frequent pattern mining algorithm respectively, and the overall time of pattern mining of all sequences was obtained. As the number of sequences increases, the overall elapsed time trends of the two algorithms are shown in Figure 7.

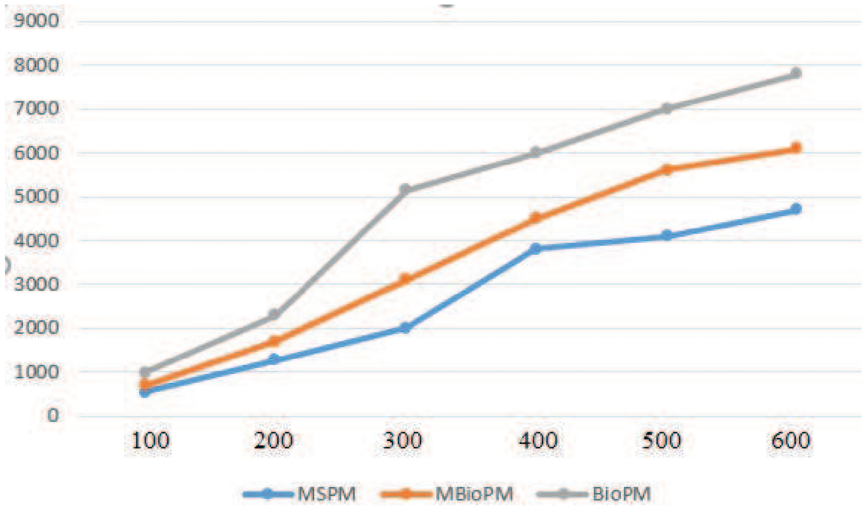


Figure 7: The relationship between the number of sequences and the time

From Figure 7, under the conditions of a certain fixed support threshold, the elapsed time of the three algorithms for comparison will increase without exception with the increase of the number of sequences. However, it can be noted that the elapsed time of the multiple sequence frequent pattern mining algorithm is more stable, especially when the model is longer, the overall time is significantly less than the other two algorithms. It is due to the fact that the BioPM algorithm needs frequent construction of the projection database during the mining process, and there are also a large number of short-term generations that cause the complexity of the algorithm's space-time and increase the efficiency. For the MBioPM algorithm, since each time a frequent pattern of k-class length is mined, the existing pattern needs to be compared with the buffer pattern one by one. At the same time, the buffer area needs to be cleared and opened repeatedly during the pattern growth. These will definitely reduce the speed of the algorithm. The multiple sequence frequent pattern mining algorithm starts frequent pattern mining from the basic pattern, avoids the runtime overhead of generating a great quantity of short patterns, and at the same time uses basic frequent pattern prefix trees for pattern growth after fast mining to obtain basic frequent pattern string sets to avoid "interference patterns". The generation of the mining efficiency of the algorithm has been greatly improved.

6 Conclusion

According to the characteristics of biological sequence patterns and mining requirements, fast and effective biological single sequence and multiple sequence pattern mining algorithms are proposed in this paper, respectively. First of all, the concept of the basic pattern of the sequence is defined. The algorithm can connect the basic patterns of the sequence to obtain the basic pattern table of the sequence. For multiple sequences, the basic pattern table of multiple sequences can be obtained through the merge operation. Based on the basic model, overall basic frequent patterns could be easily detected. In order to mine general frequent patterns, a basic frequent pattern prefix tree is constructed based on a single or multiple sequence basic frequent pattern table, and the general frequent pattern mining is quickly implemented using the set vector operation, which improves the mining efficiency. At the same time, the use of pruning technology avoids producing a great quantity of unnecessary short patterns. The analysis results of the experiment show that the two proposed algorithms significantly improve the mining efficiency of the biological sequence model, especially can get better mining results in the case of a small support threshold.

Funding

This research was supported by the Natural Science Foundation of Hebei Province (H2019201378).

Bibliography

[1] Apostolico, A.; Preparata, F. P. (1983). Optimal off-line detection of repetitions in a string, *Theoretical Computer Science*, 22(3), 297-315, 1983.

[2] Ben-Hur, A.; Brutlag, D. (2003). Remote homology detection: A motif based approach, *Bioinformatics*, 19 Suppl 1(1), 26-33, 2003.

[3] Benkaddour, M. K., Bounoua, A. (2017). Feature extraction and classification using deep convolutional neural networks, PCA and SVC for face recognition, *Traitement du Signal*, 34(1-2), 77-91, 2017.

[4] Cardon, L. R.; Stormo, G. D. (1992). Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments, *Journal of Molecular Biology*, 223(1), 159-170, 1992.

[5] Chen, L.; Liu, W. (2013). Frequent patterns mining in multiple biological sequences, *Computers in Biology & Medicine*, 43(10), 1444-1451, 2013.

[6] Crochemore, M.; Ilie, L. (2008). Maximal repetitions in strings, *Journal of Computer & System Sciences*, 74(5), 796-807, 2008.

[7] Dong, T. (2017). Assessment of data reliability of wireless sensor network for bioinformatics, *International Journal Bioautomation*, 21(1), 241-250, 2017.

[8] Jiang, Q.; Li, S.; Guo, S. (2011). A new model for finding approximate tandem repeats in DNA sequences, *Journal of Software*, 6(3), 386-394, 2011.

[9] Karmaker, S.; Ruhi, F. Y.; Mallick U. K. (2018). Mathematical analysis of a model on guava for biological pest control, *Mathematical Modelling of Engineering Problems*, 5(4), 427-440, 2018.

[10] Kurtz, S.; Choudhuri, J. V.; Ohlebusch, E.; Schleiermacher, C.; Stoye, J.; Giegerich, R. (2001). REPuter: the manifold applications of repeat analysis on a genomic scale, *Nucleic Acids Research*, 29(22), 4633-4642, 2001.

[11] Lawrence, C. E.; Reilly, A. A. (1990). An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences, *Proteins Structure Function & Bioinformatics*, 7(1), 41-51, 1990.

[12] Liao, C. C.; Chen, M. S. (2014). DFSP: A Depth-First SPelling algorithm for sequential pattern mining of biological sequences, *Knowledge* & *Information Systems*, 38(3), 623-639, 2014.

[13] Makalowski, W. (2003). Not junk after all, *Science*, 300(5623), 1246-1247, 2003.

[14] Pasquier, N.; Bastide, Y.; Taouil, R.; Lakhal, L. (1999). Efficient mining of association rules using closed itemset lattices, *Information Systems*, 24(1), 25-46, 1999.

[15] Rubin, E. M.; Lucas, S.; Richardson, P. (2004). Finishing the euchromatic sequence of the human genome, *Nature*, 431(7011), 931-945, 2004.

[16] Saqhai Maroof, M. A.; Yang, G. P.; Biyashev, R. M.; Maughan, P. J.; Zhang, Q. (1996). Analysis of the barley and rice genomes by comparative RFLP linkage mapping, *Theoretical & Applied Genetics*, 92(5), 541-551, 1996.

[17] Shapiro, J. A.; Von, S. R. (2005). Why repetitive DNA is essential to genome function, *Biological Reviews*, 80(2), 227-250, 2005.

[18] Stansfield, W.; King, R.; Mulligan, P. (2006). A dictionary of genetics, *Yale Journal of Biology & Medicine*, 75(4), 236-245, 2006.

[19] Won, J. I.; Park, S.; Yoon, J. H.; Kim, S. W. (2006). An efficient approach for sequence matching in large DNA databases, *Journal of Information Science*, 32(1), 88-104, 2006.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 590-607, August 2019.

A Factored Similarity Model with Trust and Social Influence for Top-N Recommendation

X.F. Zhang, X.L. Chen, D.W. Seng, X.J. Fang

Xuefeng Zhang
School of Computer Science and Technology, Hangzhou Dianzi University
Hangzhou 310018, China

Xiuli Chen
School of Computer Science and Technology, Hangzhou Dianzi University
Hangzhou 310018, China

Dewen Seng*
School of Computer Science and Technology, Hangzhou Dianzi University
Hangzhou 310018, China
*Corresponding author: sengdw@163.com

Xujian Fang
School of Computer Science and Technology, Hangzhou Dianzi University
Hangzhou 310018, China

Abstract: Many trust-aware recommendation systems have emerged to overcome the problem of data sparsity, which bottlenecks the performance of traditional Collaborative Filtering (CF) recommendation algorithms. However, these systems most rely on the binary social network information, failing to consider the variety of trust values between users. To make up for the defect, this paper designs a novel Top-N recommendation model based on trust and social influence, in which the most influential users are determined by the Improved Structural Holes (ISH) method. Specifically, the features in Matrix Factorization (MF) were configured by deep learning rather than random initialization, which has a negative impact on prediction of item rating. In addition, a trust measurement model was created to quantify the strength of implicit trust. The experimental result shows that our approach can solve the adverse impacts of data sparsity and enhance the recommendation accuracy.

Keywords: recommendation system, matrix factorization, trust, social influence, deep learning, top-n recommendation.

1 Introduction

The dawn of the big data era has brought abundant information resources. However, human beings are sometimes provided with too much information, that is, faced with information overload. To solve the problem, many portals and e-commerce systems have adopted recommendation systems to push the desired information to users. These systems learn users' preferences from their historical behaviors, aiming to recommend the items meeting their interests and needs.

CF is one of the most popular and successful personalized recommendation algorithms. The algorithm discovers the preferences of users by mining the data on their historical behaviors, divides the users into different groups based on their preferences, and recommends similar items to users in each group. The recommendation accuracy of the CF hinges on two issues: inferring user preferences from historical data, and identifying similar users or items. The traditional CF algorithm faces two major challenges, namely, data sparsity and cold start. Many methods have been developed to cope with these challenges. Some scholars have introduced various auxiliary data to improve the CF, including membership [7, 26], trust [2, 8, 14, 23, 27, 28] and other social information [5, 21, 25].

With the development of social networking, it is increasingly convenient to acquire the data on social relationship. Against this background, a series of trust-based approaches have been proposed and applied in various domains, making up an integral part of recommendation algorithms. Trust relationship, closely related to social information, is a hotspot in the research of recommendation systems. In social networks, users always prefer the items recommended by those they trust. Therefore, trust-aware recommendation can greatly improve the recommendation effect of sparse users (i.e. data sparsity), leading to better user experience and enhanced loyalty. Based on trust propagation mechanism, Jamali and Ester [10] designed a MF-based model for recommendation in social rating networks (SocialMF). The potential feature vectors of target users were described as the weighted average of their trustee's feature vectors, which significantly reduced the recommendation error, especially for cold-start users. Guo et al. [8] extended the SVD++ algorithm [14] into the TrustSVD algorithm, which achieves excellent recommendation effect through considering the explicit trust relationship of the target user. The above studies fully demonstrate the promotion effect of the trust relationship on recommendation accuracy. However, these trust relationships are single and incomplete.

The observations on real-world datasets like FilmTrust, Epinions and Ciao reveal two problems of trust relationship: (1) the trust relationship may be implicit due to privacy concerns, resulting in the sparsity of trust information; (2) the trust value is usually binary, i.e. each trusted friend has the same impact on the target user, which goes against the reality.

Therefore, the existing trust information can not fully mine the implicit user preference information in social networks. It is necessary to accurately measure the trust relationship, and examine its role in item recommendation.

The user behavior in the social network is also greatly affected by social influence. The behavior or opinion of a person is easily affected by influential users, who seem to be authoritative. To study user behavior, it is both necessary and difficult to analyze the influence of different users out of the massive information in the social network.

The social influence analysis [18] refers to the evaluation of the impact of each user against his/her information or social behavior by a certain standard, with the aim to identify those with significant impact in his/her group or the social network.

In fact, the analysis of social influence has attracted much attention from scholars engaging in sociology, management, information science, and economics [1,12,13,15,22]. For example, Kitsak et al. [13] empirically investigated social networks and email networks, and divided network nodes by k-kernel decomposition to different levels from the edge to the core, revealing that the most influential nodes are not necessarily the most connected nodes, i.e. those with high betweenness centrality, but those with high k-shells. Nevertheless, in the field of recommendation system, trust plays an important role, and social influence is neglected by most researchers, although it is also a key factor affecting the behavior of social network users.

Our research is mainly motivated by two issues. On the one hand, the existing trust-based recommendation methods usually use the binary trust relationship of social networks directly to improve the recommendation quality, seldom considering the difference and potential impact of trust intensity among users, and also can not fully tap the implicit user preference information in social networks, which can not achieve good recommendation effect in the case of sparse trust data.

On the other hand, the existing recommendation systems rarely utilize social influence, despite its significant impact on user behavior in the social network. To overcome these two defects, this paper fully excavates the implicit social relationships among users in the social network, and propose a top-N item recommendation algorithm which fusion of trust and social influence, called

Factored user and item Similarity Model with Trust and social Influence based on Deep learning (FSTID). The architecture of our recommendation algorithm is shown in Fig. 1. Finally, the proposed algorithm was proved more accurate and effective than other recommendation methods on three datasets (Epinions, Ciao and FilmTrust).

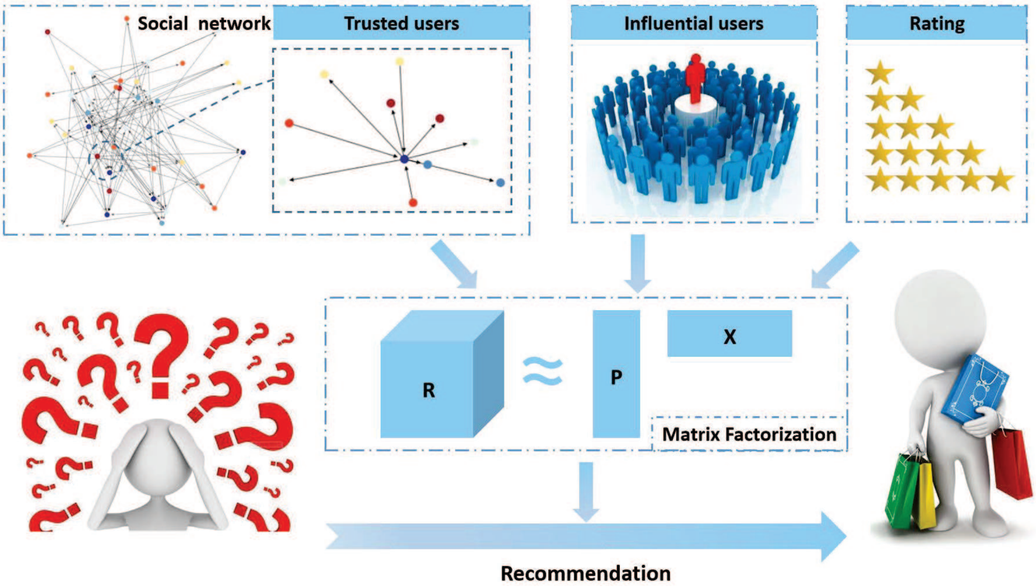


Figure 1: The architecture our recommendation algorithm

There are four main contributions of this research:

(1) A novel trust measurement model was conducted to quantify the implicit trust in social network. The trust value covers three aspects, namely, user interaction, item rating and user preference. The combination of the two kinds of relationship between trust and similarity are compact, which are effective to solve the extreme data sparsity of recommendation system. In addition, our model has played a significant positive role correspond to accurate positioning neighbor users.

(2) The ISH method was proposed to pinpoint the key nodes, i.e. the most influential users, in the social network.

(3) Considering the influence of user trust and social influence on recommendation, an innovative top-N recommendation model with the incorporation of user and item similarity, trust and social influence is proposed. Besides, Deep Autoencoder (DAE) technology was employed to optimize the initial values of the latent features for MF.

(4) Many top-N recommendation algorithms were compared through experiments on three real-world datasets. The comparison shows that our approach can alleviate the sparse problem of rating data to a certain extent and obtain more reliable recommendations.

The remainder of this paper is organized as follows: Section 2 details the construction of our algorithm; Section 3 compares our algorithm with other top-N recommendation algorithms through experiments; Section 4 puts forward the conclusions and foretells the future research.

2 Model construction

This section introduces the overall framework of the FSTID and details the four key stages of the method in seven steps. Before presenting the FSTID, a new trust measurement model was first established, the ISH method was employed to identify the most influential nodes in the social network, and the DAE was adopted to extract user and item feature vectors.

Let $U = \{u_1, u_2, \cdots, u_m\}$ be the set of m users, $I = \{i_1, i_2, \cdots, i_n\}$ be the set of n items, $R = [r_{u,i}]_{m\times n}$ be the binary matrix of the item ratings by the user ($r_{u,i} = 1$ means user u has purchased or rated item i; otherwise, user u has not purchased or rated item), $G = (V, E)$ be the graph of social network (V is the set all nodes (users) and E is the set of all edges), and $< i, j >$ be the edge from node i to node j, that is, the trust relationship between users i and j.

2.1 Overall framework

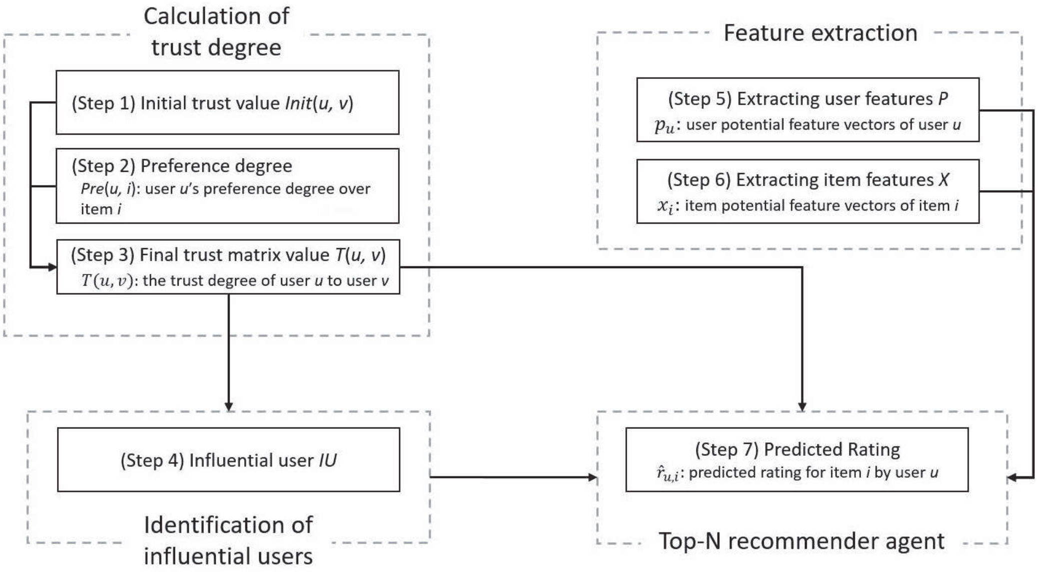

Calculation of trust degree
(Step 1) Initial trust value Init(u, v)
(Step 2) Preference degree
Pre(u, i): user u's preference degree over item i
(Step 3) Final trust matrix value T(u, v)
T(u, v): the trust degree of user u to user v
Feature extraction
(Step 5) Extracting user features P
p_u: user potential feature vectors of user u
(Step 6) Extracting item features X
x_i: item potential feature vectors of item i
(Step 4) Influential user IU
Identification of influential users
(Step 7) Predicted Rating
$\hat{r}_{u,i}$: predicted rating for item i by user u
Top-N recommender agent

Figure 2: The overall framework of the FSTID

As shown in Fig. 2, the procedure of the FSTID algorithm roughly contains four key stages: the calculation of trust value, the identification of influential users, feature extraction, and top-N recommendation. Firstly, the trust value is computed based on the initial trust value and preference degree of each user, forming the trust network. Then, the ISH is adopted to identify the key nodes in the social network, and allocate them into the set of influential users. After that, the features of users and items are extracted by the DAE.

Finally, the CF is applied to predict the item ratings, and provide personalized recommendation. The four stages are implemented in seven steps:

(1) *Calculation of trust value*

Step 1: The trust value of each user is initialized based on his/her interaction information with others. The term interaction is defined as two users rating on the same item i.

Step 2: Considering the mutual influence between trust value and interactions, and the users' tendency to trust those who have interacted successfully on their favorite items, the author measured the user preference over different items by preference degree.

Step 3: Based on user preference over item i, different weights are assigned to items in each interaction, then generate the final trust value, and the trust network is formed after filtering out the weak trust relationship.

(2) *Identification of influential users*

Step 4: The constraint value of each node is computed according to the topology of its neighbors in the directed trust network. Taking the constraint value as the indicator, the importance of each node is evaluated and recorded as "influence". Finally, the most influential users are determined and allocated to the same set.

(3) *Feature extraction*

Steps 5 and 6: By using DAE to learn users' behaviors without supervision so that the high-dimensional, sparse users' behaviors can be compressed into low-dimensional, dense users and items feature vectors. Compared to initial features, these features are no longer sparse, but also more representative.

(4) *Top-N recommendation*

Step 7: Combining the user similarities and item similarities, we introduce trust users and influential users to predict rating and offer personalized recommendation for users

2.2 Calculation of trust value (Steps 1 3)

The trust value can be expressed as $T = [t_{u,v}]_{m\times m}$, where $t_{u,v}$ is a nonzero element indicating the existence of social relationship between users u and v. In the real world, the trust relationship of most online social networks, e.g. Facebook, Epinions and Flixster, is usually represented in binary form (0 or 1), which is not exactly the same as the trust relationship of users.

This subsection sets up a novel trust measurement model, regardless if the trust value obeys explicit distribution. First, two assumptions were put forward: (1) an interaction is defined as two users perform a rating on the same item; (2) the trust value comes from cumulative experience of subjective individuals. Under these assumptions, the trust value can be initialized as:

$$Init(u,v) = \frac{min(|I_u \bigcap I_v|, D_u)}{D_u} \tag{1}$$

where $I_u \bigcap I_v$ is the number of interactions between users u and v, i.e. the number of items rated by both users; $D_u = \sqrt{|I_u|}$ is an adjustable threshold specifying the minimum number of interactions between the two users that fully trust each other. The trust relationship is not a stable factor, which along with influence of interaction experience will be changed. A successful interaction would increase the trust value and an unsuccessful interaction act oppositely. If both users u and v have rated item i, then the interaction is successful if the two ratings are both above or below the average rating of user himself:

$$\begin{cases} successful, & (r_{u,i} - \bar{u}) * (r_{v,i} - \bar{v}) \geqslant 0 \\ unsuccessful, & (r_{u,i} - \bar{u}) * (r_{v,i} - \bar{v}) < 0 \end{cases} \tag{2}$$

where $\bar{u}$ and $\bar{v}$ are the average rating of users u and v on all items, respectively.

The trust value differs with the preference degree of each user. Let $Pre(u,i)$ be the preference degree of user u over item i, U_i be the set of users that have rated item i, and o be a user in the set U_i. Then, the similarity between users u and o can be calculated using the Pearson correlation coefficient (PCC):

$$sim(u,o) = \left(\frac{1}{2} + \frac{\sum_{i\in I_u \cap I_o} (r_{u,i} - \bar{u})(r_{o,i} - \bar{o})}{2 * \sqrt{\sum_{i\in I_u \cap I_o} (r_{u,i} - \bar{u})^2} \sqrt{\sum_{i\in I_u \cap I_o} (r_{o,i} - \bar{o})^2}} \right) * \frac{|I_u \cap I_o|}{|I_u|} \tag{3}$$

$$Pre(u,i) = \frac{\sum_{o\in U_i} sim(u,o)}{|U_i|} \tag{4}$$

Based on the user preferences in successful or unsuccessful interactions, different weights were assigned to different items. On this basis, the final trust value $T(u,v)$ can be obtained as:

$$T(u,v)=\frac{\sum_{i\in successful}Pre(u,i)-\sum_{i\in unsuccessful}Pre(u,i)}{\sum_{i\in successful}Pre(u,i)+\sum_{i\in unsuccessful}Pre(u,i)}*Init(u,v) \tag{5}$$

2.3 Identification of influential nodes (Step 4)

In the social network, each user collects and disseminates information as a node. Some of them are more influential than others. These key nodes are similar to Structural Hole (SH) occupants in the competitive group theory proposed by Burt [3]. The SH occupants play a key role in information exchange between the local groups and enjoys more opportunities and options than other group members. Therefore, the author improved the SH method to integrate the impacts of in- and out- degrees of neighbor nodes in the directed trust network. The ISH is illustrated with an instance in Fig. 3 and Table 1. Here, the in-degree is defined as the number of nodes pointing towards a node, and the out-degree has the opposite meaning.

The SH is generally evaluated by two indices: the betweenness centrality and the index given by Burt himself. The former refers to Freeman's betweenness centrality [6] for the overall network and its extended form. If a node is on the shortest path of many pairs of other nodes, then the node has a high betweenness centrality, and a high probability to be a SH occupant. The latter involves four aspects: effective size, efficiency, constraint and hierarchy. Constraints refer to the node's ability to use SH in its own networks, which is more important than the other three aspects. The calculation formula of constraint is as follows, describing the degree to which a node in a network is directly or indirectly connected to other nodes:

$$C(i)=\sum_{j\in\Gamma(i)}\left(p(j,i)+\sum_{q\in\Gamma(i)}p(j,q)*p(q,i)\right)^2,i\neq q\neq j \tag{6}$$

where $\Gamma(i)$ refers to the nodes directly connected to node i in the undirected graph; $p(j,i)$ is the proportion of the energy that node i devotes to maintain its relationship with node j to the total energy of node i. Since the trust network is a directed graph, $\Gamma(i)$ was replaced with T_i^- to describe the set of nodes pointing towards node i. The value of $p(j,i)$ can be calculated by:

$$p(j,i)=\frac{Z_{ji}}{\sum_{j\in T_i^-}Z_{ji}} \tag{7}$$

where $Z_{ji}=1$ if $<i,j>\in E$ (otherwise, $Z_{ji}=0$); $p(j,i)+\sum_{q\in\Gamma(i)}p(j,q)*p(q,i)$ represents the time and energy that node i invests directly or indirectly to maintain its connection with node j, when node j is the only node pointing to node i, the item reaches a maximum value of 1; this item reaches a minimum when there is no bridge node between node i and node j.

Based on the $C(i)$ value obtained from Eq. (6), the number and closeness of neighbors can be evaluated in a comprehensive manner. The $C(i)$ value is negatively correlated with the number of nodes pointing towards the target node, and positively with the closeness between these nodes. If closely distributed, the nodes are less likely to acquire new information. On the contrary, nodes with small constraint coefficients can promote information propagation.

Table 1 lists the constraint values of all nodes in Fig.3 (a), respectively calculated by the SH and the ISH. During the calculation, any node with no node pointing towards it was neglected. As shown in the table, nodes u_4 and u_7 have the same constraint value, i.e. the same influence. The information flow theory [20] holds that the information on the Internet often propagates to the opinion leader, before spreading to the others distributed in a wide range. In this sense,

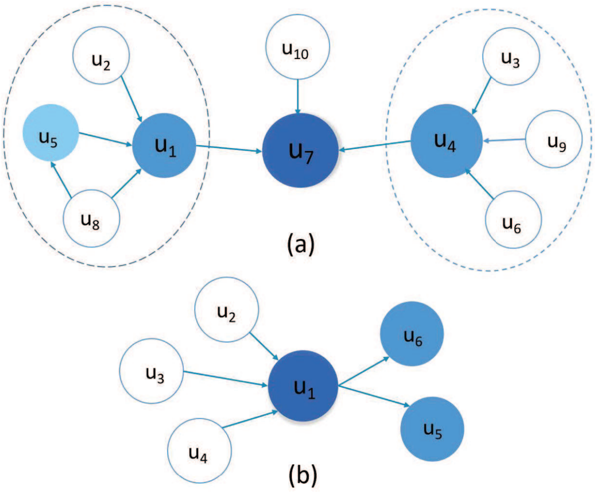


Figure 3: Two toy examples of trust network topological graph

Table 1: Constraint values of all nodes in Fig. 3(a) obtained by the SH and the ISH

$C(i)$	u_1	u_4	u_7	u_8
SH	0.6667	0.3333	0.3333	1
ISH	0.5555	0.3333	0.0625	1

a node connected to multiple opinion leaders is likely to become an SH occupant. Taking the in-degree of node as the evaluation standard for opinion leader, nodes u_1, u_4 and u_7 with the highest in-degrees must be opinion leaders, and play a greater role than the other nodes (e.g. u_8 and u_9) in information dissemination. Among them, node u_7 are connected to two opinion leaders at the same time, indicating that it is more conducive to information dissemination than nodes u_1 and u_4. In other words, node u_7 is more likely to occupy an SH than the other two nodes. The SH results in Table 1 are obviously unreasonable, as many important nodes are not found. This is because the traditional SH only measures the relationship between a node and its nearest neighbor, without considering that between the node and its two-hop neighbors.

For convenience, it is assumed that a key node j can boost the influence of the node i it points towards. The nodes pointing away from node j were also taken into account. As shown in Fig.3(b), the influence of node u_6 is reduced due to the fact that the only node pointing towards it u_1 has a node pointing away from it u_5. Therefore, the ISH integrating the impacts of in- and out-degrees of neighbor nodes can be expressed as:

$$C(i) = \sum_{j \in T_i^-} \frac{|T_j^+|}{|T_j^+| + |T_j^-|} * \left(p(j,i) + \sum_{q \in T_i^-} p(j,q) * p(q,i) \right)^2, i \neq q \neq j \tag{8}$$

where T_j^+ is the set of nodes pointing away from node j.

The validity of the ISH was tested with the instance in Fig.3(a). The test results are recorded in Table 1. It can be seen that node u_7, with the lowest constraint value, is the most influential node. Both u_1 and u_4 had an in-degree of 3, yet u_8, a node pointing towards u_1, has a connection with u_5, which reduces the influence of u_1 on u_8. Thus, the influence of u_4 is higher than that of u_1.

After the constraint values of all nodes were obtained by Eq. (8), the top $k\%$ of users with the smallest constraint values were selected as global influential users IU.

2.4 Feature extraction (step 5 and step 6)

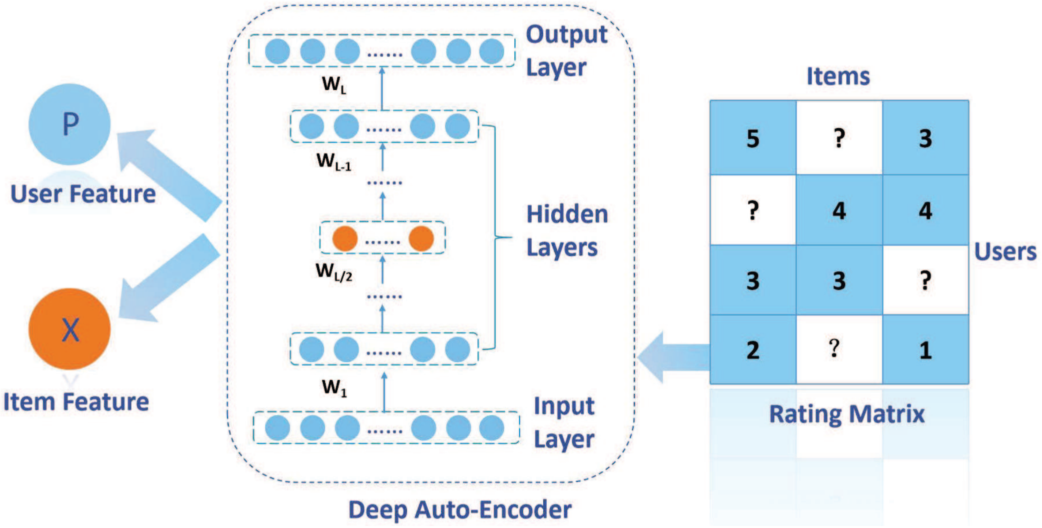


Figure 4: The flowchart of feature extraction.

Considering the sheer number of users and items, the user vectors and item vectors were respectively mapped by the MF to d-dimensional joint potential factor spaces $P_{d\times m}$ and $X_{d\times n}$. Explain intuitively that matrix P represents the preferences of m users on d topics, and X represents the d topics which are concerned by users behind n items, then the rating matrix R can be approximately expressed as the product of two matrices: $R = P^T X$. Most MF-based recommendations initialize P and X randomly and iteratively output local optimal solutions, which depends heavily on the initial values [4]. This paper attempts to make more accurate recommendations by mining out accurate feature vectors from the implicit features of the MF. To this end, the DAE, a deep neural network, was adopted to compress high-dimensional, sparse user behaviors into low-dimensional, dense feature vectors of users and items. This approach provides good initial values to the network through unsupervised layer-by-layer pre-training, and adjust the network weights by supervised fine-tuning training, making it possible to extract key information from the data and form features [24].

The primary goal is to design an item-based (user-based) DAE to map the observed $U_i(I_u)$ into a low-dimensional potential (implicit) space, and then reconstruct $U_i(I_u)$ in the output space to learn advanced, abstract features. The entire workflow is illustrated in Fig.4. Let $S = [I_1, ..., I_i, ..., I_m]$ be the input. The specific steps of the DAE to extract user feature matrix P are given below:

Step 1: Given an input S, let $W_i \in \mathbb{R}^{d\times m}(i \in [1, L-1])$ be a weight matrix, $b_i \in \mathbb{R}^d$ be the bias vectors and $\sigma(x) = \frac{1}{1+e^x}$ be the sigmoid formula. Then, the d-dimensional implicit feature h_i can be expressed as:

$$h_1 = \sigma(W_1 S + b_1) \tag{9}$$

$$h_i = \sigma(W_i h_{i-1} + b_i) \tag{10}$$

Step 2: Given the weight matrix $W_L \in \mathbb{R}^{m\times d}$ between hidden and output layers, and bias vector $b_L \in \mathbb{R}^m$, the DAE can reconstruct the original data $\hat{S}$ from the hidden layer h_{L-1} by:

$$\hat{S} = \sigma(W_L h_{L-1} + b_L) \tag{11}$$

Step 3: The pretraining program of the DAE intends to minimize the objective function by adjusting weight matrices W and bias vectors b:

$$L = \frac{1}{2m} \sum_{i=1}^{m} \left| \hat{S}_i - S_i \right|^2 + \frac{\lambda}{2} |W_1|^2 + \ldots + \frac{\lambda}{2} |W_L|^2 \tag{12}$$

where $\hat{S}_i$ and S_i represent the i-dimensional vectors of $\hat{S}$ and S, respectively; $|\cdot|^2$ is the squared L2 norm of vectors or matrices, i.e. the sum of squared dimensional values; the first factor in objective function denotes the error which is employed to minimize the error of reconstructed data $\hat{S}_i$ and original data S_i; the other factors are regular terms used to prevent overfitting.

Step 4: Update weight matrix W in each iteration by:

$$W = W - l \times \frac{\partial L}{\partial W} \tag{13}$$

where l refers to learning rate. According to the above formula, bias vectors b update process act in the same way.

The d-dimensional feature vector of the users and the items were obtained by repeating the above steps to continuously optimize the model until the end of training.

2.5 Top-N recommendation agent (step 7)

Our approach FSTID, is based on the model proposed by Guo called Factored user and item Similarity model with social Trust (FST) [9], which predict user u's rating on item i through four parts: (1) item bias b_i; (2) the similarity between user u and another user v who has rated item i: $p_v^T q_u$, where p_v, q_u are the implicit feature vectors of users v and u, respectively; (3) the similarity between item i and another item j which has been rated by user u: $x_j^T y_i$, where x_j and y_i are the implicit feature vectors of items j and i, respectively; (4) influence of any user u's trusted user w on targeted item i: $p_w^T y_i$. The rating prediction formula can be expressed as:

$$\hat{r}_{u,i} = s|U_{i-u}|^{-\beta} \sum_{v \in U_{i-u}} p_v^T q_u + (1-s)|I_{u-i}|^{-\alpha} \sum_{j \in I_{u-i}} x_j^T y_i + |T_u|^{-z} \sum_{w \in T_u} p_w^T y_i + b_i \tag{14}$$

where U_{i-u} refers to the set of users having rated item i except user u; I_{u-i} refers to the set of items having been rated by user u except item i; T_u is the set of users trusted by user u; $s \in [0,1]$ is the relative importance of user similarity; α, β, z are parameters related to the number of rated items, similar users and trusted users, respectively. Taking β for instance, $\beta = 1$ refers to the mean user similarity, item i will get a high rating only if all users in U_{i-u} are similar to user u; when $\beta = 0$, this item calculates the sum of the similarities between user u and user in U_{i-u}, which means that item i will get a high rating even if only a few users are similar to user u. In conclusion, the probability of obtaining a high predictive rating decreases with the increase of parameter β.

The above analysis shows the critical importance of the influential users' ratings on item i. In addition, Guo et al. [9] confirmed that the trustee of user u also affects user u's rating on item i. Thus, the rating prediction formula of the FST can be modified as:

$$\begin{aligned} \hat{r}_{u,i} =& s|U_{i-u}|^{-\beta} \sum_{v \in U_{i-u}} p_v^T q_u + (1-s)|I_{u-i}|^{-\alpha} \sum_{j \in I_{u-i}} x_j^T y_i + \delta |T_u^+|^{-z} \sum_{o \in T_u^+} p_o^T y_i + \\ & (1-\delta)|T_u^-|^{-z} \sum_{c \in T_u^-} p_c^T y_i + |IU|^{-\mu} \sum_{f \in IU} p_f^T y_i + b_i \end{aligned} \tag{15}$$

The trustees in the FST were divided into a set of trustees T_u^+ and a set of trustors T_u^-, aiming to disclose their impacts on the rating of item i. The parameter $\delta \in [0,1]$ was adopted to control the weights of the two types of users. When $\delta = 0$, it represents the influence of trustees is totally ignored. To the contrary, $\delta = 1$ denotes only the influence of trustees is considered. $\mu \geq 0$ refers to the parameter of the number of influential users. For each influential user $f \in IU$, its influence over item i is described by the inner product $p_f^T y_i$. In addition, the model parameters b, P, Q, X and Y can be trained by minimizing the objective function below:

$$\begin{aligned} J = & \frac{1}{2} \sum_{u \in U} \sum_{i \in I_u^+, j \in I_u^-} |(r_{u,i} - r_{u,j}) - (\hat{r}_{u,i} - \hat{r}_{u,j})|_F^2 + \\ & \frac{\lambda}{2} (|P|_F^2 + |Q|_F^2 + |X|_F^2 + |Y|_F^2 + |b|_F^2) \end{aligned} \tag{16}$$

where U is the set of all users; I_u^+ and I_u^- respectively refers to the rated items and unrated items of user u. For simplicity, parameter λ was taken as the regularization term in all cases. The pseudocode for model learning is shown in Algorithm 0, where α, β, z, μ, s and δ are control parameters, λ is the regularization parameter, η is the initial learning rate and ρ is the number of samples. Firstly, ρ unrated items are selected randomly for each user to train the model (Line 6).

Algorithm 1 The learning algorithm of FSTID

```
    Data:
    $U \leftarrow$ user set; $R \leftarrow$ user item matrix;$T \leftarrow$ user trust matrix;
    $IU \leftarrow$ influential user set; $P \leftarrow$ user feature matrix; $X \leftarrow$ item feature matrix;
    Input:
    $\alpha$, $\beta$, $z$, $\mu$, $s$, $\delta$, $\rho$, $\lambda$, $\eta$ and iteration limitation $L$;
    Output:
    $b$, $P$, $Q$, $X$, $Y$

 1: Initialize bias vector $b$ with random values in (0, 0.01), and set $Q = P, Y = X$;
 2: while $J$ not converged or the number of iterations $< L$ do
 3:     for all $u \in U$ do
 4:         for all $i \in I_u^+$ do
 5:             $Z \leftarrow sample(\rho, I_u^-)$
 6:             Compute Loss $J$ by Eq. (16)
 7:             Update $b_i, b_j, q_u, y_i, y_j, p_v, x_k, p_o, p_c, p_f$ according SGD:
 8:             $b_i \leftarrow b_i - \eta \times \frac{\partial J}{\partial b_i}$,   $q_u \leftarrow q_u - \eta \times \frac{\partial J}{\partial q_u}$,   $y_i \leftarrow y_i - \eta \times \frac{\partial J}{\partial y_i}$
 9:             for all $j \in Z$ do
10:                 $b_j \leftarrow b_j - \eta \times \frac{\partial J}{\partial b_j}$,   $y_j \leftarrow y_j - \eta \times \frac{\partial J}{\partial y_j}$
11:                 $\forall v \in U_{j-u}, p_v \leftarrow p_v - \eta \times \frac{\partial J}{\partial p_v}$
12:             end for
13:             $\forall v \in U_{i-u}, p_v \leftarrow p_v - \eta \times \frac{\partial J}{\partial p_v}$,   $\forall k \in U_{u-i}, x_k \leftarrow x_k - \eta \times \frac{\partial J}{\partial x_k}$
14:             $\forall o \in T_u^+, p_o \leftarrow p_o - \eta \times \frac{\partial J}{\partial p_o}$,   $\forall c \in T_u^-, p_c \leftarrow p_c - \eta \times \frac{\partial J}{\partial p_c}$
15:             $\forall f \in IU, p_f \leftarrow p_f - \eta \times \frac{\partial J}{\partial p_f}$
16:         end for
17:     end for
18:     iteration number++
19: end while
20: return $b$, $P$, $Q$, $X$, $Y$
```

Then, the parameters are optimized continuously in the training phase by Stochastic Gradient Descent (SGD) (Lines 7-15) until the loss function converges or the preset maximum number of iterations is reached (Line 3). Finally, the learning vectors and matrices are outputted (Line 20). As for comparative experiments, we will make detailed comparisons and explanations in subsequent experiments to express the advantages of our algorithm.

3 Experiments and results analysis

This section carries out a series of experiments on three real-world datasets, trying to answer the following questions:

(1) Does the ISH enjoy higher accuracy in identifying influential users than other influence measurement methods?

(2) How do model parameters like $\alpha,\ \beta,\ z$ and μ affect recommendation accuracy?

(3) How does the number of influential users affect recommendation accuracy?

(4) Does the FSTID outperform the other advanced trust-aware recommendation algorithms on typical sparse datasets?

3.1 Datasets and evaluation metrics

There real-world datasets were selected for our experiments, including FilmTrust, Ciao and Epinions, which are currently well-known test data sets. Most trust-based recommendation algorithms are used to test the performance of algorithms because they contain both user ratings and trust relationships. FilmTrust is a dataset from a movie sharing website. The data entries in the dataset are movie ratings based on a four-point scale. Ciao and Epinions are datasets from two famous consumer review websites, on which users rate commodities against a five-point scale and refer to others' ratings before deciding on whether to purchase a commodity. As shown in Table 2, all the three datasets are extremely sparse in nature.

Table 2: Specification of the used datasets

Data Set	Users	Items	Ratings	Density
Epinions	40163	139738	664824	0.0118%
Ciao	7375	99746	139738	0.0379%
FilmTrust	1508	2071	40163	1.14%

*Density=$\frac{\#Ratings}{\#Users\times\#Items}$

The 5-fold cross validation was adopted in our experiments. Each dataset was split randomly into 5 equal parts. In each iteration, 1 part was selected as the test set and the other 4 parts as the training set. The mean results of the 5 parts were taken as the final results. Different from the evaluation of rating prediction, the top-N recommendation performance was measured by two metrics: precision and F1-measure. F1-measure is the weighted average of precision and recall rate (the proportion of all recommended items in the items rated by the target user). Let U' be the set of users in the test set and $R_N(u)$ be the top-N items recommended to user u. Then, the precision, recall rate and F1-measure under N recommended items, respectively denoted as $P@N$, $R@N$ and $F1@N$, can be calculated by:

$$P@N = \frac{1}{|U'|}\sum_{u\in U'}\frac{|R_N(u)\cap I_u|}{N} \tag{17}$$

$$R@N = \frac{1}{|U'|}\sum_{u\in U'}\frac{|R_N(u)\cap I_u|}{I_u} \tag{18}$$

$$F1@N = \frac{2*P@N*R@N}{P@N+R@N} \tag{19}$$

where $N \in \{5, 10\}$ is the number of items recommended to the target user. The greater the values of P@N and F1@N, the more accurate the top-N recommendation.

3.2 Social influence analysis

The Susceptible Infected (SI) epidemic model [15] was selected to evaluate the effects of model parameters on recommendation results. Mimicking the virus propagation process, this classic model is a mature method to simulate the transmission of information. In this model, each network node either exists in the susceptible S status or in the infected I status. Once infected, a susceptible node will irreversibly become an infected node, and transmit virus to its neighbor nodes at the probability of $\gamma \in (0,1)$. In our experiments, the probability γ was set to 0.001 and the network nodes were initialized randomly. The influence of each network node on the three datasets were evaluated by the parameter S_i at transmission time $t = 10$. The mean value of S_i can be calculated by:

$$\bar{S_i} = \frac{1}{M}\sum_{m=1}^{M} S_i \tag{20}$$

where M is the repeated times of node i.

Our method ISH was compared with the SH and the Degree Centrality (DC) in terms of the social influence over the three datasets. The experimental results are presented in Fig.5, where the x-axis is the calculated influence of each node and the y-axis is the actual influence of each node.

The purple area in Fig.5 are low-impact nodes. The SH and the DC had almost the same number of low-impact nodes on all three datasets. Comparatively, the SH nodes ranked lower than the DC nodes, indicating that the SH enjoyed better effect than the DC. Meanwhile, the ISH achieved the lowest proportion of low-impact nodes among the three methods. Most ISH nodes concentrated in the second half of the ranking.

The red area in Fig.5 are high-impact nodes. By analysis of top-25 nodes, the number of high-impact nodes calculated by DC, SH, and ISH are 18, 15 and 18 in FilmTrust; the number of high-impact nodes is 8, 9, 11 respectively in Ciao and also is 9, 8, 10 in Epinions. The results are very close, revealing that the node with higher degree is more important.

As shown in Fig.5, the ISH achieved the highest mean value on all datasets, that is, our algorithm outperformed the other methods in identification. To sum up, the ISH can identify the top $k\%$ influential users correctly, without mistaking low-impact users as high-impact users.

3.3 Effects of model parameters

Effect of parameters $\alpha,\ \beta,\ z,\ \mu$

The parameters $\alpha,\ \beta,\ z,\ \mu$ respectively control the influence of item similarity, user similarity, trustees, and influential users on the item ratings. To save time and space, the values of

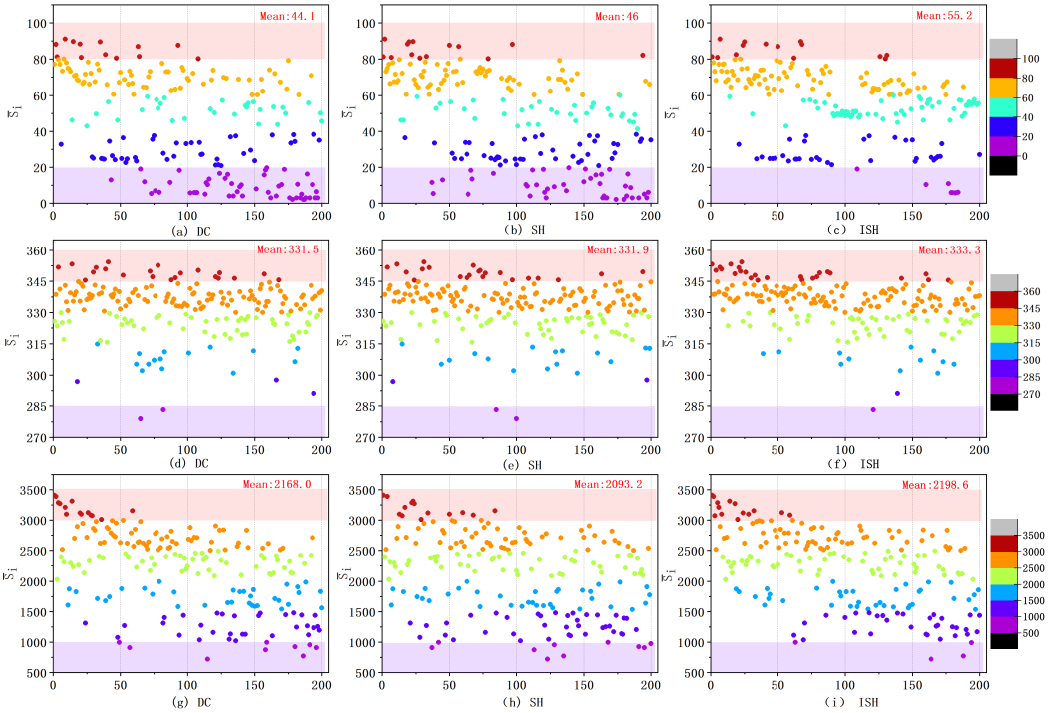


Figure 5: Correlation analysis of different methods and their actual influence. (a)-(c) Results on FilmTrust; (d)-(f) Results on Ciao; (g)-(i) Results on Epinions.

these parameters were limited to a small set: 0.5, 1, 2 while the values of s and δ were fixed at 0.5 in our experiments. For simplicity, only the five best P@10 results on each dataset are listed in Table 3. Obviously, the different parameter configurations led to different results, and the optimal parameters changed from dataset to dataset. The optimal values of α, β, z and μ on Epinions, Ciao and FilmTrust were (0.5, 2, 2, 0.5), (0.5, 2, 0.5, 0.5) and (0.5, 1, 0.5, 0.5), respectively. It can be concluded that the optimal combination is $\alpha = 0.5$, β, $z > 1$ and $\mu < 1$, that is, the item similarity and influential users are more important than user similarity and trustees.

Table 3: The five best P@10 results on each dataset.

	Epinions	Ciao	FilmTrust
1	0.010486 (0.5, 2, 2, 0.5)	0.02378 (0.5, 2, 2, 2)	0.352258 (2, 1, 1, 0.5)
2	0.010467 (1, 2, 0.5, 0.5)	0.023609 (0.5, 0.5, 0.5, 0.5)	0.351964 (2, 0.5, 2, 2)
3	0.010379 (2, 1, 2, 0.5)	0.023411 (1, 2, 1, 1)	0.351948 (0.5, 1, 0.5, 1)
4	0.010243 (1, 2, 1, 1)	0.023396 (0.5, 2, 1, 0.5)	0.351819 (1, 1, 0.5, 0.5)
5	0.010231 (0.5, 2, 0.5, 2)	0.02336 (2, 0.5, 0.5, 1)	0.351775 (0.5, 0.5, 0.5, 0.5)

Effect of s and δ

The parameters s and δ from Eq.(15) respectively describe the relative importance of user similarity and trustees on the prediction of item ratings. The value of s is positively correlated with the impact of user similarity, while the value of δ is negatively correlated with the impact of trustees. In our experiments, α, β, z, μ were set to the optimal values, while the values of

s and δ were increased at a step of 0.1 in the range [0, 1]. Firstly, the value of s was set to 0.5 to obtain a series of results on δ and determine the value of δ. Then, the experiments were conducted with s. According to the experimental results in Fig. 6, the value of P@10 was worse than most other values at $\delta = 0$ or $\delta = 1$. This means the recommendation accuracy can be improved effectively through the proper combination of trustors and trustees. This conclusion also applies to parameter s. The value of s can be set to 0.3, despite the difference in its optimal value on different datasets.

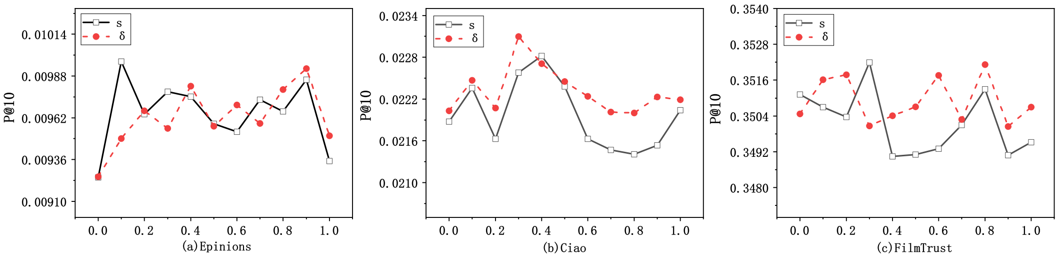


Figure 6: The effects of s and δ on FSTID at P@10

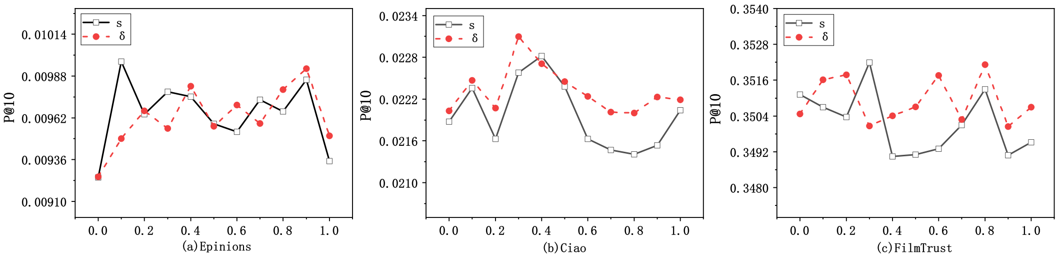


Figure 7: The effects of parameters k on FSTID at P@10

Effect of parameter k

The FSTID takes the top k% users with higher influence as influential users, as only a tiny fraction of users in the social network have a great influence. To verify the impact of parameter k on the recommendation performance, the value of the parameter was increased at the step of 1 in the interval [0, 10]. The experimental results are displayed in Fig. 7. It can be seen that the optimal value of k was 2, 4 and 7 in Epinions, Ciao, and FilmTrust, respectively. Moreover, the poorest result was observed on the three datasets at $k = 0$, i.e. ignoring influential users. Hence, the consideration of influential users can indeed enhance the recommendation effect.

3.4 Comparison with other methods

Several top-N recommendation algorithms were selected to evaluate the effect of our method, such as MostPop, FST, Group Bayesian Personalized Ranking (GBPR), the factored item similarity model (FISM), and the FSTID-. The MostPop is the baseline method that ranks the ratings of an item by popularity, i.e. how frequently the item is rated or consumed by the user. Proposed by Guo et al., the FST [8] takes account of implicit user feedback, similarity and social trust. The GBPR was improved from the Bayesian Personalized Ranking (BPR) by Pan and Chen [16], which introduces the influence of social groups on user preference to enhance the item recommendation quality. FISM is a top-N recommendation method based on item similarity proposed by Kabbur et al. [11]. The FSTID- removes the DAE pretraining from our approach and randomly sets the values of P, Q, X and Y in the interval of (0, 0.01).

Table 4: Results of different methods at N=5 and 10. The optimal results are in bold characters.

Datasets	N	Metrics	MostPop	GBPR	FISM	FST	FSTID-	FSTID
Epinions	5	P@N	0.01169	0.009353	0.01147	0.01179	0.011888	**0.01231**
	10	F1@N	0.01298	0.01103	0.01307	0.0133	0.013107	**0.01402**
	5	P@N	0.009171	0.00756	0.00902	0.009187	0.009486	**0.01024**
	10	F1@N	0.01305	0.01111	0.01315	0.01328	0.014431	**0.01459**
Ciao	5	P@N	0.02677	0.02228	0.02704	0.02741	0.027382	**0.0283**
	10	F1@N	0.02436	0.02063	0.02495	0.02523	0.025442	**0.02644**
	5	P@N	0.02142	0.01827	0.02141	0.02174	0.022376	**0.02329**
	10	F1@N	0.02662	0.02116	0.02687	0.0272	0.028402	**0.02914**
FilmTrust	5	P@N	0.4170	0.4124	0.4171	0.4191	0.418567	**0.4198**
	10	F1@N	0.4095	0.4051	0.4087	0.4099	0.411364	**0.4116**
	5	P@N	0.3503	0.3470	0.3503	0.3514	0.352159	**0.3532**
	10	F1@N	0.4518	0.4458	0.4516	0.4521	0.452825	**0.4541**

It can be seen from Table 4 that the FSTID achieved better results than the contrastive methods on all datasets, proving that our approach can alleviate data sparsity and make reliable recommendations. Besides, the FSTID, which relies on the DAE to initialize the MF, outperformed the FSTID-, revealing that the features extracted by the DAE are more suitable than randomly generated features. On the other hand, FSTID- because there is no need to utilize DAE for feature initialization, it has lower computational cost than FSTID, and in most experiments it has achieved the best results compared with other approach except the FSTID method, which indicates the effectiveness of the integration of social influences to improve the recommended performance. Furthermore, the FSTID's advantage over the other methods were smaller on FilmTrust than Ciao and Epinions, due to the relatively small scale of FilmTrust. This means the DAE can capture features more effectively in big data.

4 Conclusion

This paper proposes a novel factored similarity model for top-N recommendation based on trust and social influence. Firstly, we quantify the trust value between users based on the user's interaction information and similarity, and finally build the user's trust association graph. In the process of calculating the trust degree, the penalty factor based on the common rating item is added to reduce the accidental effect caused by the low number of common rating items, which greatly increases the attack cost of malicious users, makes it difficult to become the neighbors of target users, and effectively enhances the stability of the algorithm. Then, the ISH was utilized to identify the influential users in a complex network. Moreover, the DAE was adopted to compress high-dimensional, sparse user behaviors into low-dimensional, dense vectors of user and item features. Compared with other top-N recommendation approaches, our approach achieved excellent recommendation effects on three real-world datasets. The future research will consider the dynamic changes in user preference and trust with the time of user rating, and try to recommend items preferred by users in a timely manner.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 61473108), the Fundamental Public Welfare Research Program of Zhejiang Province (No. GF19F020052) and the National Experimental Teaching Demonstration Center of Computer-Science in China.

Bibliography

[1] Anagnostopoulos, A.; Kumar, R.; Mahdian, M. (2008). Influence and correlation in social networks, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 7–15, 2008.

[2] Bao, Y.; Fang, H.; Zhang, J. (2014). Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation, *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 350: 30–36, 2014.

[3] Burt, R.S. (2009); *Structural holes: The social structure of competition*, Harvard university press, 2009.

[4] Deng, S.; Huang, L.; Xu, G. et al. (2017). On deep learning for trust-aware recommendations in social networks, *IEEE Transactions on Neural Networks* & *Learning Systems*, 28(5), 1164–1177, 2017.

[5] Deng, X.Y.; Wang, C. (2018). A hybrid collaborative filtering model with context and folksonomy for social recommendation, *Ingenierie des Systemes d'Information*, 23(5), 139–157, 2018.

[6] Freeman, L.C. (1977); A set of measures of centrality based on betweenness, *Sociometry*, 40(1), 35–41, 1977.

[7] Guy, I.; Ronen, I.; Wilcox, E. (2009). Do you know?: recommending people to invite into your social network, *Proceedings of the 14th International Conference on Intelligent User Interfaces*, 77–86, 2009.

[8] Guo, G.; Zhang, J.; Yorke-Smith, N. (2016). A novel recommendation model regularized with user trust and item ratings, *IEEE Transactions on Knowledge* & *Data Engineering*, 28(7), 1607–1620, 2016.

[9] Guo, G.; Zhang, J.; Zhu, F. et al. (2017). Factored similarity models with social trust for top-n item recommendation, *Knowledge-Based Systems*, 122, 17-25, 2017.

[10] Jamali, M.; Ester; M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks, *Proceedings of the fourth ACM conference on Recommender systems*, 135–142, 2010.

[11] Kabbur, S.; Ning, X.; Karypis, G. (2013). Fism: factored item similarity models for top-n recommender systems, *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 659-667, 2013.

[12] Kempe, D.; Kleinberg, J.; Tardos, E. (2003). Maximizing the spread of influence through a social network, *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 137–146, 2003.

[13] Kitsak, M.; Gallos, L.K.; Havlin, S. et al. (2010). Identification of influential spreaders in complex networks, *Nature Physics*, 6(11), 888–893, 2010.

[14] Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 426–434, 2008.

[15] Li, D.; Luo, Z.; Ding, Y. et al. (2017). User-level microblogging recommendation incorporating social influence, *Journal of the Association for Information Science and Technology*, 68(3), 553-568, 2017.

[16] Pan, W.; Chen, L. (2013). Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering, *Proceedings of The Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 13, 2691-2697, 2013.

[17] Pastorsatorras, R.; Castellano, C.; Mieghem, P.V. et al. (2014). Epidemic processes in complex networks, *Review of Modern Physics*, 87(3), 120-131, 2014.

[18] Peng, S.; Wang, G.; Xie, D. (2017). Social influence analysis in social networking big data: Opportunities and challenges, *IEEE Network the Magazine of Global Internetworking*, 31(1), 11–17, 2017.

[19] Rendle, S.; Freudenthaler, C.; Gantner, Z. et al.. (2009). Bpr: Bayesian personalized ranking from implicit feedback, *Conference on Uncertainty in Artificial Intelligence*, 452-461, 2009.

[20] Rogers, E.M. (1995). *The Diffusion of Innovations*, Free Press, 1995.

[21] Sedhain, S.; Menon, A.K.; Sanner, S. et al. (2017). Low-rank linear cold-start recommendation from social data, *Proceedings of the 31th AAAI Conference on Artificial Intelligence (AAAI)*, 1502–1508, 2017.

[22] Xu, W.; Rezvani, M.; Liang, W. et al. (2017). Efficient algorithms for the identification of top-k structural hole spanners in large social networks, *IEEE Transactions on Knowledge & Data Engineering*, 29(5), 1017-1030, 2017.

[23] Xu, K.; Zheng, X.; Cai, Y. et al. (2018). Improving user recommendation by extracting social topics and interest topics of users in unidirectional social networks, *Knowledge Based Systems*, 140, 120–133, 2018.

[24] Yuan, X.; Huang, B.; Wang, Y. et al. (2018). Deep learning based feature representation and its application for soft sensor modeling with variable-wise weighted SAE, *IEEE Transactions on Industrial Informatics*, (99): 1–1, 2018.

[25] Yang, C. ; Sun, M. ; Zhao; W. X. et al. (2016). A neural network approach to joint modeling social networks and mobile trajectories, *Acm Transactions on Information Systems*, 35(4), 36, 2016.

[26] Yuan, Q.; Zhao, S.; Chen, L. et al. (2009). Augmenting collaborative recommender by fusing explicit social relationships, *Workshop on Recommender Systems and the Social Web*, Recsys, 2009.

[27] Zhang, Z., Liu, Y., Jin, Z. et al. (2018). A dynamic trust based two-layer neighbor selection scheme towards online recommender systems, *Neurocomputing*, 285, 94-103, 2018.

[28] Zhao, H.; Yao, Q.; Kwok, J.T. et al. (2017). Collaborative filtering with social local models, *2017 IEEE International Conference on Data Mining (ICDM)*, 645–654, 2017.

Appendix

In order to facilitate the reader to read better, we have listed the abbreviations used in this article in lexicographic order, as shown in Table 5.

Table 5: Acronyms and Initialisms Dictionary.

Abbreviation	Full name
CF	Collaborative Filtering
DAE	Deep Autoencoder
DC	Degree Centrality
FISM	Factored Item Similarity Model
FST	Factored user and item Similarity model with social Trust
FSTID	Factored user and item Similarity Model with Trust and social Influence based on Deep learning
GBPR	Group Bayesian Personalized Ranking
ISH	Improved Structural Holes
MF	Matrix Factorization
SGD	Stochastic Gradient Descent
SH	Structural Hole
SI	Susceptible Infected

Author index