文章编号:2096-1472(2025)10-0042-05
DOI:10.19644/j.cnki.issn2096-1472.2025.010.009

基于 BLS 签名算法的 PBFT 共识算法研究

邵 琪[1],刘从军[1,2]

(1.江苏科技大学计算机学院,江苏 镇江 212000;
2.江苏科大汇峰科技有限公司,江苏 镇江 212000)
✉ 15370096752@163.com; liu_cj@163.com

摘 要:为针对应用于区块链中的 PBFT 共识算法存在计算效率低、不适用于节点数量庞大的区块链系统、扩展性差等问题,基于 BLS 聚合签名算法,设计了 BLS-PBFT 共识算法。针对 PBFT 共识过程中两次全部节点之间的相互广播机制,使用 BLS 签名算法将各节点之间的广播改为主节点对各节点的单播,在原 Prepare 阶段各节点的签名通过验证之后,对这些签名进行聚合操作。经实验对比,在 BLS-PBFT 算法的共识过程中,通信开销降低了约 85%。在节点数量为 19 个时,共识时延降低了约 80%,吞吐量提高了近 2.5 倍,证明了本算法的有效性和可行性。

关键词:联盟链;PBFT;BLS 签名算法
中图分类号:TP301.6 **文献标志码**:A

Research on PBFT Consensus Algorithm Based on BLS Signature Algorithm

SHAO Qi[1], LIU Congjun[1,2]

(1.*School of Computer, Jiangsu University of Science and Technology, Zhenjiang* 212000, *China*;
2.*Jangsu KEDAHUIFENG Technology Co., Ltd., Zhenjiang* 212000, *China*)
✉ 15370096752@163.com; liu_cj@163.com

Abstract: To address the issues of low computational efficiency, poor scalability, and unsuitability for large-scale blockchain systems in the PBFT consensus algorithm applied to blockchain, this paper proposes the BLS-PBFT consensus algorithm based on the BLS aggregate signature scheme. By replacing the two rounds of all-to-all broadcasting in the original PBFT process with a unicast mechanism based on BLS signatures, the algorithm converts node broadcasts into unicast communication from the primary node. Signatures from nodes in the Prepare phase are aggregated after verification. Experimental comparison demonstrates that the BLS-PBFT algorithm reduces communication overhead by approximately 85%. With 19 nodes, consensus latency decreases by about 80%, while throughput increases by nearly 2.5 times, validating the algorithm's effectiveness and feasibility.

Key words: consortium blockchain; PBFT; BLS signature algorithm

0 引言(Introduction)

共识算法是区块链网络的核心技术之一,其主要功能是确保网络中各个节点对交易信息和区块达成一致。在数据共识的过程中会遇到很多问题,如节点故障、节点宕机、网络通信异常以及恶意节点攻击等。共识算法的职责就是通过协议来协调各节点之间的行为,确保系统的正常运行。

收稿日期:2024-12-23

基于联盟链中的应用大都依赖于PBFT(Practical Byzantine Fault Tolerance)共识算法。当联盟链规模扩大,节点数增多时,通信量会呈平方级增长,导致信息通信阻塞,进而降低整个系统的工作效率。

针对上述问题,本文基于BLS聚合签名算法,优化PBFT算法的Prepare和Commit阶段,将共识过程中存在的两次全部节点的广播优化为主节点对各从节点的单播行为,在不影响共识结果的情况下,极大降低了共识过程中的通信复杂度。

1 相关理论(Relevant theory)

1.1 区块链技术

区块链[1]技术的提出最早可以追溯到2008年,名为中本聪的匿名作者发表了一篇题为《比特币:一种点对点的电子现金系统》的白皮书。该白皮书描述了一种去中心化的电子货币,即比特币,而区块链作为比特币的底层技术,逐渐被一些学者和比特币爱好者了解及研究[2]。自此,区块链作为一种分布式、去中心化的技术,凭借其去透明性、不可篡改、安全性高等优势,应用于能源、司法、物联网、房地产等领域。随着区块链的快速发展,对区块链节点体量的要求越来越大,对共识机制的要求也越来越高[3],因此,共识机制的可扩展性、安全性和稳定性就成为了区块链领域的重点研究问题[4]。

根据访问权限、控制、去中心化程度、隐私及区块节点准入规则的不同,区块链被划分为公有链、私有链、联盟链3种类型。公有链是一种完全去开放的区块链,任何节点都可以访问、参与和验证区块的交易和数据[5];私有链的权限受到严格限制,只有特定的参与方或者经过授权的用户才可以参与其中;联盟链的开放程度介于公有链和私有链之间,由特定的组织或实体共同管理。相比于公有链,联盟链的去中心化程度较低;但相比于私有链,其并不是由个体完全控制,而是由成员组织节点共同管理控制。3种区块链的对比详见表1。

表1　区块链的对比表

Tab.1 Comparison table of blockchains

区块链	去中心化程度	隐私性	工作效率	应用场景
公有链	高	低	低	数字货币、开放应用平台
私有链	低	高	高	企业内部、特定应用
联盟链	较低	较高	较高	企业间合作、行业间合作

1.2 PBFT共识算法

1999年MONRAT等[4]为了提升BFT算法的效率,提出PBFT共识算法。使其时间复杂度从$O(n^{f+1})$(f为网络中拜占庭节点的数量)降为$O(n^2)$。假设在一个分布式网络中共存在N个节点,其中,恶意节点数量为f,PBFT算法能容忍错误节点<节点总数1/3,即两者满足$N\geqslant 3f+1$时,全网能够对消息达成共识[6]。

PBFT共识流程分为5个阶段,当主节点发生宕机,就会触发视图切换协议,切换主节点继续完成整个流程。PBFT算法的共识过程为

(1)Request阶段:客户端向主节点发送消息。

(2)Pre-Prepare阶段:客户端的请求消息送达至主节点后,由主节点将生成的Pre-Prepare消息广播给其他所有的从节点。

(3)Prepare阶段:通过验证后的Pre-Prepare消息,从节点生成Prepare消息,并将消息进行全网广播,同时接收其他从节点的消息。

(4)Commit阶段:当各节点接收的有效Prepare消息$F>2f+1$(f为网络中拜占庭节点的数量)时生成Commit消息,并将消息进行全网广播,同时接收其他从节点的消息。

(5)Reply阶段:当各节点接收的Commit消息$F>2f+1$(f为网络中拜占庭节点的数量)时生成Reply消息,并返回给客户端。当客户端收到的Reply消息数量超过f时,表示该消息已通过共识。共识流程如图1所示。

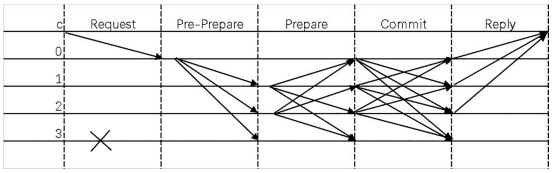

图1　PBFT共识算法流程图

Fig.1 PBFT consensus algorithm flow chart

1.3 BLS签名算法

2004年BONEH等[7]提出BLS签名算法。BLS签名算法的基础是椭圆线性密码学。相较于传统的数字签名算法,BLS签名算法将多个不同节点的私钥聚合成一个短签名,该短签名即为聚合公钥,此方法减少了节点间的信息交换次数。

BLS签名修改了哈希算法,将最后的哈希结果映射到椭圆曲线上的一个点,将未签名的消息散布到该点后,使用双线性映射函数的交换特质,在私钥没有泄露的前提下,进行签名的验证[8]。

BLS签名的具体步骤为

(1)初始化:在$G0$,$G1$,GA循环群中定义一个函数e,如式(1)所示:

$$e:G0\times G1\rightarrow GA \tag{1}$$

(2)生成密钥:将一个随机生成的数r作为密钥,公钥即为$pk=r\times G$。

(3)签名:根据双线性映射函数的性质,将未签名的消息m映射到椭圆曲线上对应的点$E(m)$,经过计算生成签名$s=r\times E(m)$。

(4)验证:只需判断签名是否满足验证等式,如式(2)所示。其中,P为公钥,H是消息m映射到G上的点:

$$e(P,H)=e(G,s) \tag{2}$$

之所以能实现这样的验证,是因为函数e有以下的特殊性质,如式(3)~式(5)所示,其中,x、a、b都是取自有限域Zp的标量。

$$e(P,Q+R)=E(P,Q)\times E(P,R) \tag{3}$$

$$e(x\times P,Q)=e(P,x\times Q) \tag{4}$$

$$e(a\times P,b\times Q)=e(P,Q)^{a\times b} \tag{5}$$

综上所述,假设聚合签名为$S=S0+S1+S2$,聚合公钥为$P=P0+P1+P2$,推导得出等式成立,其中p是循环群的阶的素数如式(6)所示:

$$\begin{aligned}e(G,S)&=e(G,S0+S1+S2)\\&=e(G,(pk0+pk1+pk2)\times H)\end{aligned}$$

$$=e((pk0+pk1+pk2)\times G,H)$$
$$=e(P0+P1+P2,H)$$
$$=e(P,H) \quad (6)$$

2 BLS-PBFT 算法模型(BLS-PBFT algorithm model)

BLS-PBFT 利用聚合签名技术优化了原 PBFT 机制中的 Prepare 和 Commit 阶段的广播通信行为,并且优化了 Request 阶段的广播通信行为。

Prepare 阶段和 Commit 阶段的具体优化操作如下:首先,各个节点使用私钥对主节点发送来的消息进行签名后再返回给主节点;然后,由主节点对从节点发来的签名消息进行验证,并将所有签名消息进行聚合生成 Commit 消息,发送给其他从节点;最后,其他从节点利用 BLS 签名算法生成的聚合公钥对签名进行验证。

Request 阶段的优化如下:传统 PBFT 算法可能存在主节点利用主导地位故意转发虚假信息,为了避免出现此类情况,客户端会将请求消息发送给 $2f+1$ 个(f 为网络中拜占庭节点的数量)节点。BLS-PBFT 算法的共识过程为

(1)Request 阶段:由客户端向 $2f+1$ 个节点(其中包含主节点)发送请求消息,根据请求消息,主节点生成 Pre-Prepare 消息。同时,其他节点启动计时器,当主节点出现宕机、超时情况,启动视图切换协议,及时更换主节点。

(2)Pre-Prepare 阶段:主节点利用 BLS 签名算法在 Pre-Prepare 消息中加入签名,然后将生成的消息发送给其他的从节点。

(3)Prepare 阶段:主节点将预准备消息送达至各从节点后,由从节点使用主节点公钥对消息签名验证,验证如式(2)所示。然后,收到客户端请求消息的从节点会将请求消息与收到的预准备信息进行验证,比对失败时会向其他节点发送视图切换消息,更换主节点。验证通过后,从节点使用私钥对预准备消息进行签名,并将消息发送回主节点。

(4)Commit 阶段:当各从节点的准备消息送至主节点后,用对应从节点的公钥验证准备消息的签名。所有验证都完成后会计算主节点收到的准备消息的数量 F,当 $F>2f+1$(f 为网络中拜占庭节点的数量)后,主节点对 Prepare 消息聚合签名并生成确认消息。

(5)Reply 阶段:确认消息发送至各从节点后,生成聚合公钥并验证签名。验证如式(6)所示。聚合签名通过验证后会,各节点将回复消息广播至客户端节点。客户端计算回复消息的数量 F,当 $F>f+1$ 时,表明该条消息完成共识流程。

BLS-PBFT 共识算法如图 2 所示。

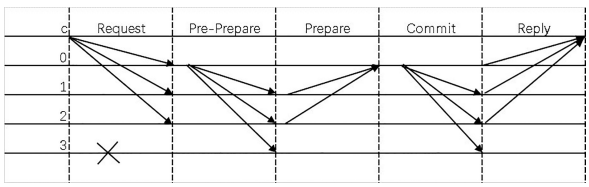

图 2 BLS-PBFT 共识算法流程图

Fig. 2 BLS-PBFT consensus algorithm flow chart

综上所述,经过计算得出 BLS-PBFT 算法的通信复杂度为 $O(n)$,计算详情见式(7)。而 PBFT 算法的通信复杂度为 $O(n^2)$,两者相比,BLS-PBFT 算法的通信复杂度更低,有更高的工作效率。在节点数量多的场景中,BLS-PBFT 更具优势。

$$M_{\text{BLS-PBFT}}=(2/3n+1)+(n-1)+(n-1)+(n-1)+n$$
$$=14/3n-2 \quad (7)$$

主节点处理流程如图 3 所示。

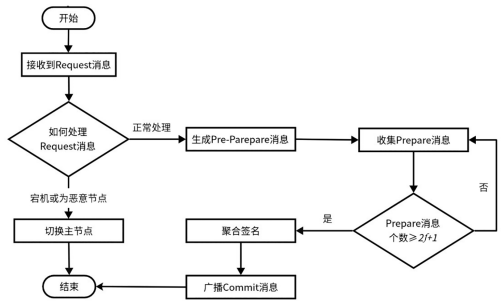

图 3 主节点共识过程

Fig. 3 Master node consensus process

从节点的共识过程如图 4 所示。

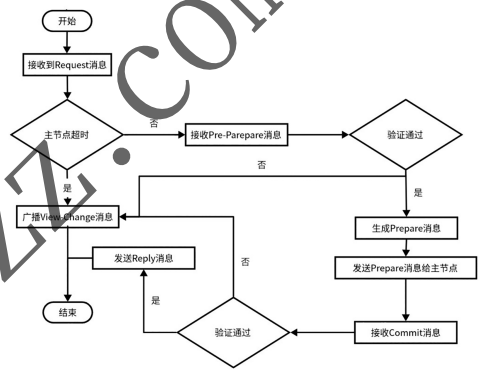

图 4 从节点共识过程

Fig. 4 Backup node consensus process

3 实验设计及结果分析(Experimental design and result analysis)

为了测试改进后算法的性能提升程度,基于 Golang 和 Python 语言以及 TCP 的 P2P 通信实现节点间通信,使用密码学函数库 PBC 中的 BLS 签名算法进行实验。本章从签名算法的耗时、通信量、共识时耗和吞吐量方面进行实验,配置信息详见表 2。

表2 配置信息

Tab. 2 Configuration information

名称	配置
CPU	AMD Ryzen 75 800H
内存	16 GB
操作系统	CentOS 7.6
语言	Golang,Python

3.1 BLS 聚合签名耗时

相比原 PBFT 算法,改进后的算法增加了聚合签名的操作。本文通过实验得出 BLS 签名所花费的时间,实验结果如图 5 所示。本次实验测试以 300 次聚合签名为基础,按照每百次签名为单位逐渐增加至 1000 次签名,进行 1000 次聚合签名

的耗时接近0.5ms。原算法在Prepare阶段和Commit阶段需要多次节点间通信,聚合签名的耗时远小于节点间通信的耗时。

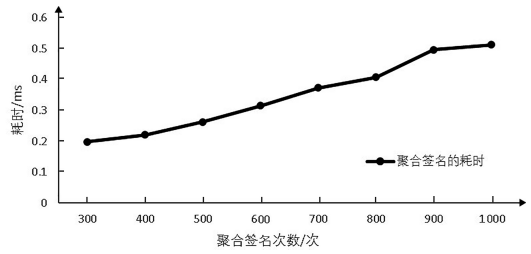

图5　聚合签名耗时

Fig. 5 Time consumption of signature aggregation

3.2　通信量对比

通信量开销很大程度上影响区块链规模大小,节点之间通信量过大可能会导致网络时延增加、数据丢失等问题。本次实验基于Shell脚本语言,编写脚本控制启动节点的数量。

不同大小的区块会对区块链中交易处理速度、费用、吞吐量等多方面造成不同程度的影响。因此,在本次实验过程中,需保证在相同大小的区块上进行实验。本次实验设定共识消息为1 MB/次,通信量的计算如式(8)所示:

$$T=M\times Ms \tag{8}$$

通过实验与计算得出PBFT与BLS-PBFT的通信量开销,实验结果如图6所示。

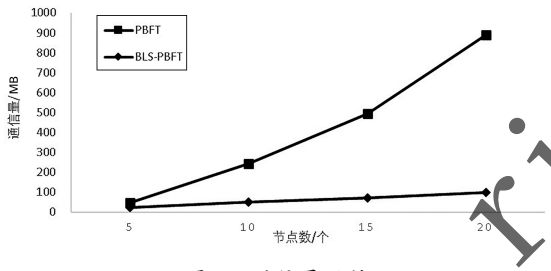

图6　通信量开销

Fig. 6 Communication overhead

3.3　共识过程耗时对比

共识过程耗时是以客户端向其他节点发送消息开始,直到客户端收到$\geqslant 2f+1$条确认消息为止,这个过程消耗的时间。为了计算共识过程的耗时,在客户端发送给主节点的请求消息上会增加一条时间戳记录作为开始时间,在客户端收到的第$2f+1$条回复消息上增加一条时间戳作为结束时间,两者相减得出本次共识所耗费的时长。本次实验设定的节点数量分别为4、7、10、13、16、19个,在不同节点数量下进行5次实验,计算5次实验的平均值为耗时结果。实验结果如图7所示。

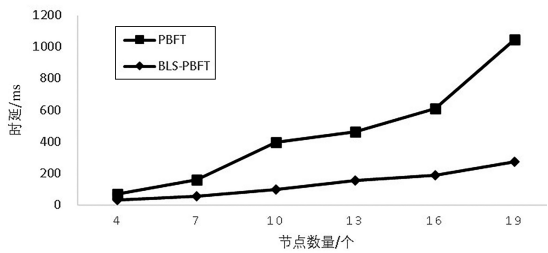

图7　共识时延对比

Fig. 7 Consensus latency comparison

图7的实验结果表明,当节点数量较少时,两种算法的共识过程耗时相差较小。当系统中节点数量增加时,BLS-PBFT共识算法耗时的增长速度明显低于PBFT共识算法。当节点数量为19时,PBFT的时延接近1000 ms,而BLS-PBFT的时延不到200 ms,两者相比,时延降低了近80%。实验结果表明,BLS-PBFT算法在节点数量增多的情况下,仍保持较低的网络时延。

3.4　吞吐量对比

吞吐量(TPS)是指在单位时间内能够成功处理的数据量或请求数。它是衡量算法性能的重要指标之一。设定节点的数量分别为4、7、10、13、16、19个,网络中每个区块的信息数量为1000条,为了确保实验数据的稳定,客户端向主节点循环发送30s的请求消息,将中间10s的平均值作为结果。最后基于Python编写的脚本分析数据,得出最终的时延结果。实验结果如图8所示。

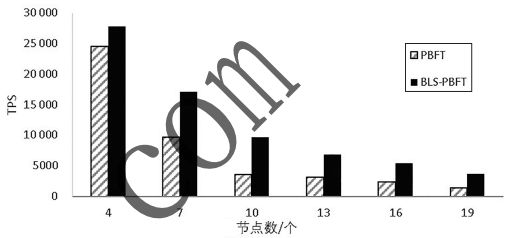

图8　吞吐量对比

Fig. 8 Throughput comparison

实验结果表明:节点数量的增加,两种算法的吞吐量都呈下降趋势,BLS-PBFT算法的吞吐量一直高于PBFT算法,且PBFT算法的下降趋势较快。当系统中节点的数量为19个时,PBFT算法的吞吐量已经低于1400,BLS-PBFT算法的吞吐量高于3600,两者相比,TPS提升了近2.5倍。实验结果再次证明了,BLS-PBFT算法具有更好的拓展性和稳定性。

4　结论(Conclusion)

基于BLS签名算法的聚合签名功能,对PBFT算法的Request、Prepare、Commit阶段进行优化,本文提出BLS-PBFT共识算法。旨在对PBFT共识算法的通信量开销、共识过程耗时、吞吐量等问题进行改进。通过实验得出BLS-PBFT算法在通信量开销、共识过程耗时、吞吐量几个方面的表现都高于PBFT共识算法。

参考文献(References)

[1] XU M,CHEN X,KOU G. A systematic review of blockchain [J]. Finacial innovation,2019,5(1):14.

[2] 陈佳伟,冼祥斌,杨振国,等. 结合BLS聚合签名改进实用拜占庭容错共识算法[J]. 计算机应用研究, 2021,38(7):1952-1955,1962.

[3] 袁昊天,李飞. 基于改进Raft共识算法和PBFT共识算法的双层共识算法[J]. 计算机应用研究, 2024,41(5):1314-1320.

[4] MONRAT A A,SCHELEN O,ANDRESSON K. A survey of blockchain from the perspectives of applications, challenges, and opportunities[J]. IEEE Access, 2019(99):1-18.

[5] 冉成科. 区块链技术在高职院校创客型师资培养中的应用探索[J]. 模具制造, 2024,24(12):72-74.

[6] 郭上铜,王瑞锦,张凤荔. 区块链技术原理与应用综述[J]. 计算机科学, 2021,48(2):271-281.

[7] BONEH D, LYNN B, SHACHAM H. Short signatures from the weil pairing[M]. International conference on the theory and application of cryptology and information security. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001:514-532.
[8] 崔彬. 云数据完整性验证技术综述[J]. 信息技术与信息化, 2020(12):8-10.

作者简介:

邵 琪(1999-),男(汉族),徐州,硕士生。研究领域:智能信息处理。
刘从军(1974-),男(汉族),武汉,高级实验师,硕士。研究领域:智能信息处理。本文通信作者。

(上接第 21 页)

[7] 孙吉贵,刘杰,赵连宇. 聚类算法研究[J]. 软件学报, 2008, 19(1):48-61.
[8] HAN D Y, KIM J. Unified Simultaneous Clustering and Feature Selection for Unlabeled and Labeled Data[J]. IEEE transactions on neural networks and learning systems, 2018,29(12):1-16.
[9] WU Z, LEAHY R. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 1993, 15(11):1101-1113.
[10] CHENG B, YANG J C, YAN S C, et al. Learning with l1-graph for image analysis[J]. IEEE transactions on image processing: a publication of the IEEE signal processing society, 2010, 19(4):858-866.
[11] ZHU X T, LOY C C, GONG S G. Constructing Robust Affinity Graphs for Spectral Clustering[C]// IEEE. IEEE Conference on Computer Vision & Pattern Recognition. Columbus: IEEE, 2014:1450-1457.
[12] NIE F P, WANG X Q, HUANG H. Clustering and projected clustering with adaptive neighbors [M]. ACM, 2014:977-986.
[13] NIE F P, WANG X Q, JORDAN M I, et al. The Constrained laplacian rank algorithm for graph-Basedclustering[C] // AAAI Press. Proceedings of the AAAI conference on artificial intelligence. Phoenix: AAAI Press. 2016, 30(1):86-102.
[14] SHI D, ZHU L, LI Y K, et al. Robust structured graph clustering[J]. IEEE transactions on neural networks and learning systems, 2019, 31(11):4424-4436.
[15] CRISTIANINI N, SHAWETAYLOR J. An introduction to support vector machines[M]. Beijing, China Machine Press, 2005:16-17.
[16] ZHOU T, THUNG K H, LIU M X, et al. Multi-modal latent space inducing ensemble SVM classifier for early dementia diagnosis with neuroimaging Data[J]. Medical image analysis, 2019, 60:101630.
[17] XU J, LIU X L, HUO Z Y, et al. Multi-Class support vector machine via maximizing multi-class margins[C] // NSF Public Access Repository(PAR). The 26th International Joint Conference on Artificial Intelligence (IJCAI 2017). Eisenhower Avenue: NSF Public Access Repository(PAR). 2017:3154-3160.
[18] NIE F P, WANG X Q, HUANG H. Multiclass capped p-Norm SVM for robust classifications[C] // AAAI. Proceedings of the AAAI conference on artificial intelligence. Phoenix: AAAI. 2017, 31(1):435-450.

作者简介:

蒋露露(2000-),女(汉族),无锡,硕士生。研究领域:图像处理和机器学习。本文通信作者。
沈项军(1977-),男(汉族),镇江,教授,博士。研究领域:模式识别与智能决策系统。

(上接第 30 页)

[10] WANG Z, ZHOU X, ZHAO W, et al. Assessing the severity of sleep apnea syndrome based on ballistocardiogram [J]. PloS one, 2017, 12(4):e0175351.
[11] Al-ANGARI H M, SAHAKIAN A V. Use of sample entropy approach to study heart rate variability in obstructive sleep apnea syndrome[J]. IEEE transactions on biomedical engineering, 2007, 54(10):1900-1904.
[12] CAO K, LV X. Multi-task feature fusion network for obstructive sleep apnea detection using single-lead ECG signal[J]. Measurement, 2022, 202:111787.

作者简介:

李轩逸(1998-),男(汉族),温州,硕士生。研究领域:机器学习与智能应用。
朱耀东(1970-),男(汉族),南通,教授,博士。研究领域:智能机器人与健康物联网。本文通信作者。
赵伟岚(1999-),男(汉族),邵阳,硕士生。研究领域:机器学习与智能应用。
樊紫岩(1998-),女(汉族),邢台,硕士生。研究领域:机器学习与智能应用。