Journal of New Technologies in Distributed Systems and Algorithmic Computing
Islamic Azad University of Sabzevar
E-ISSN: 3115-705X
https://sanad.iau.ir/journal/ntds

Research Article

Performance Comparison of Cache Algorithms in Named Data Networks Focusing on Contamination Attack Mitigation and β Parameter Optimization

Atefeh Vaez Sharestni [1] |Mohammad Reza Khayam bashi *[2] | Farmarz Safi [3]

[1]Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran, Atefehvaez@gmail.com

[2]Faculty of Computer Engineering , University of Isfahan , Isfahan , Iran, M.r.khayyambashi@eng.ui.ac.ir

[3]Big Data Research Center, Najafabad Branch, Islamic Azad University , Najafabad, Iran, Fsafi@iaun.ac.ir

Corresponding Author
***Mohammad Reza Khayam bashi**, professor, Faculty of Computer Engineering ,University of Isfahan , Isfahan , Iran, M.r.khayyambashi@eng.ui.ac.ir

Main Subject: Comparison of Cache Algorithms
Received: 5 June 2025
Accepted: 10 July 2025

https://doi.org/10.82195/ntds.2025.1208980

Abstract

Intelligent cache management in Named Data Networks (NDN) is a critical component in optimizing network performance and enhancing the quality of user experience. Tuning cache sensitivity based on behavioral changes using the β parameter and Zipf distribution is very important in cache management. Considering that β has a direct effect on the balance between resistance to pesticide contamination and keeping the real popular content, setting and determining this parameter plays a fundamental role. In this research, a new framework based on the algorithm of directed incremental Lagrange multipliers (ADMM) is presented for distributed and adaptive adjustment of β parameter. The proposed model with a decentralized structure can be implemented at the level of network nodes and leads to fast convergence and precise adjustment of β through coordinated updates of primary and secondary variables. In addition, a more advanced version called ADMM-Enhanced has been developed, which achieves more stability, accuracy and adaptability to network dynamic conditions by using the automatic adjustment of λ and ρ parameters. To evaluate the efficiency of this method, four network topologies (simple, hierarchical, advanced and mesh) were designed and simulated in different scenarios. The simulation results and comparison with the reference algorithms showed that the ADMM-Enhanced algorithm in the stable scenario increased the hit rate by 30% and reduced the contamination to zero. In the face of severe attack (30% attacker), it has shown stable, adaptable and resistant performance by maintaining CHR equal to 18% and CPR equal to 18%. Also, in complex topologies such as mesh, the proposed algorithm was able to maintain cache efficiency and quality against distributed threats, congestion and link interruption, and showed high compatibility with dynamic environments.

Keywords: ADMM Algorithm, ADMM-Enhanced, Named Data Networks, Cache Memory, Cache Hit Rate, Cache Pollution Percentage

فناوری های نوین در سیستم های توزیع شده و محاسبات الگوریتمی
دانشگاه آزاد اسلامی واحد سبزوار
E-ISSN: 3115-705X
https://sanad.iau.ir/journal/ntds

پژوهشی

مقایسه عملکرد الگوریتم‌های کَش در شبکه‌های داده‌های نام‌گذاری شده با تمرکز بر کاهش حملات آلودگی و بهینه‌سازی پارامتر β

عاطفه واعظ شهرستانی[۱] | محمدرضا خیام باشی*[۲] | فرامرز صافی[۳]

[۱]گروه مهندسی کامپیوتر، واحد نجف‌آباد، دانشگاه آزاد اسلامی، نجف‌آباد، ایران،
Atefehvaez@gmail.com

[۲]استاد، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران،
M.r.khayyambashi@eng.ui.ac.ir

[۳]مرکز تحقیقات کلان‌داده، واحد نجف‌آباد، دانشگاه آزاد اسلامی، نجف‌آباد، ایران،
Fsafi@iaun.ac.ir

نویسنده مسئول
*محمدرضا خیام باشی، استاد دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران،
M.r.khayyambashi@eng.ui.ac.ir

عنوان اصلی: عملکرد الگوریتم‌های کَش
تاریخ دریافت: ۱۵ خرداد ۱۴۰۴
تاریخ پذیرش: ۱۹ تیر ۱۴۰۴

https://doi.org/10.82195/ntds.2025.1208980

چکیده:

مدیریت هوشمند حافظه نهان در شبکه‌های داده‌های نام‌گذاری‌شده (NDN)، یک مؤلفه حیاتی در بهینه‌سازی کارایی شبکه و ارتقای کیفیت تجربه کاربر است. تنظیم حساسیت کَش بر اساس تغییرات رفتاری با استفاده از پارامتر β و توزیعZipf در مدیریت حافظه نهان بسیار مهم است. با توجه به اینکه β تأثیر مستقیمی بر تعادل میان مقاومت در برابر آلودگی کَش و حفظ محتوای واقعی محبوب دارد، تنظیم و تعیین این پارامتر نقشی بنیادین ایفا می‌کند. در این پژوهش، چارچوبی نوین مبتنی بر الگوریتم روش ضرایب لاگرانژ افزایشی جهت‌دار (ADMM) برای تنظیم توزیع‌شده و تطبیقی پارامتر β ارائه شده است. مدل پیشنهادی با ساختاری غیرمتمرکز، قابلیت اجرا در سطح گره‌های شبکه را داراست و از طریق به‌روزرسانی‌های هماهنگ متغیرهای اولیه و دوگانه، به همگرایی سریع و تنظیم دقیق β منتهی می‌شود. افزون بر آن، نسخه پیشرفته‌تری تحت عنوان ADMM-Enhancedتوسعه یافته است که با بهره‌گیری از تنظیم خودکار پارامترهای ρو λ ، پایداری، دقت و تطبیق‌پذیری بیشتری نسبت به شرایط دینامیک شبکه حاصل می‌کند. برای ارزیابی کارایی این روش، چهار توپولوژی شبکه (ساده، سلسله‌مراتبی، پیشرفته و مش) در سناریوهای مختلف طراحی و شبیه‌سازی شد. نتایج شبیه سازی و مقایسه با الگوریتم‌های مرجع نشان دادند که الگوریتم ADMM-Enhanced در سناریوی پایدار نرخ اصابت کَش را تا ۳۰٪ افزایش داده و آلودگی را به صفر رسانده است، و در مواجهه با حمله شدید (۳۰٪ مهاجم) نیز با حفظ CHR برابر ۱۸٪ و CPR برابر ۱۸٪، عملکردی پایدار، تطبیق‌پذیر و مقاوم از خود نشان داده است. همچنین در توپولوژی‌های پیچیده مانند مش، الگوریتم پیشنهادی توانست در برابر تهدیدات توزیع‌شده، ازدحام و قطع لینک، بهره‌وری و کیفیت کَش را حفظ کرده و سازگاری بالایی با محیط‌های پویا از خود نشان دهد.

کلیدواژه‌ها: الگوریتم ADMM، ADMM-Enhanced، شبکه‌های داده‌های نام‌گذاری شده، حافظه نهان، نرخ اصابت کَش، درصد آلودگی کَش

۱-مقدمه

در دهه‌های اخیر، شبکه‌های رایانه‌ای با رشد بی‌سابقه حجم داده‌ها، کاربران و تنوع کاربردهای اینترنتی مواجه شده‌اند. معماری سنتی اینترنت که بر پایه مدل میزبان‌محور و آدرس‌دهی IP طراحی شده است، در پاسخ‌گویی به نیازهای جدیدی نظیر تحویل محتوای سریع، انعطاف‌پذیر، امن و مقیاس‌پذیر، با چالش‌های متعددی روبه‌رو شده است[۱]. در پاسخ به این چالش‌ها، معماری‌های نوین با رویکرد محتوامحور[۱] (ICN) مطرح شده‌اند که در آن‌ها تمرکز اصلی از «مکان محتوا» به «نام محتوا» منتقل می‌شود [۲, ۳]. یکی از مهم‌ترین نمونه‌های معماری‌های محتوامحور، شبکه داده‌های نام‌گذاری‌شده[۲] (NDN) است. در این معماری، داده‌ها به‌صورت مستقل از مکان، نام‌گذاری شده و از طریق بسته‌های درخواستی و داده بین گره‌ها رد و بدل می‌شوند. یکی از مزایای کلیدیNDN ، بهره‌گیری از حافظه نهان بین‌راهی در تمامی گره‌هاست. این قابلیت منجر به کاهش تأخیر واکشی داده، کاهش بار سرورها، صرفه‌جویی در پهنای باند و افزایش کارایی شبکه می‌شود[۴, ۵].

با این حال، همین ویژگی مفید می‌تواند نقطه ضعف امنیتی نیز باشد. یکی از تهدیدات مهم درNDN ، حملات آلودگی حافظه نهان[۳] (CPA) است. در این نوع حملات، مهاجم با ارسال درخواست‌های مکرر برای محتوای نامحبوب یا جعلی، موجب می‌شود داده‌های پرکاربرد از کَش خارج شده و فضای کَش با داده‌های کم‌ارزش پر شود. این فرآیند به کاهش نرخ اصابت کَش، افت کارایی کلی شبکه و افزایش تأخیر پاسخ‌گویی می‌انجامد. بسیاری از سیاست‌های کَش سنتی مانند کمترین استفاده اخیر LRU [۶][۴] و کمترین استفاده شده به‌طور مکرر[۵] LFU ، به دلیل عدم توجه به الگوهای رفتاری مهاجمین یا منشأ درخواست‌ها، در برابر این نوع حملات آسیب‌پذیر هستند[۷, ۸].

در سال‌های اخیر، الگوریتم‌هایی با هوشمندی بالاتر طراحی شده‌اند تا با حملات آلودگی حافظه نهان مقابله کنند. یکی از نوآورانه‌ترین این الگوریتم‌ها، الگوریتم محبوبیت به‌ازای هر ورودی[۶] (PFP) است که با نرمال‌سازی سهم محبوبیت بین مسیرهای ورودی مختلف، از تمرکز مصنوعی محبوبیت توسط مهاجم جلوگیری می‌کند[۹]. نسخه پیشرفته‌تر این الگوریتم، PFPبا کهنگی پویا(PFP-DA) ، عامل زمان و میزان استفاده مجدد را نیز در تصمیم‌گیری دخیل می‌سازد[۱۰, ۱۱]. همچنین، نسخه بهینه‌شده این الگوریتم با استفاده از پارامتر(PFP-β) β ، سعی دارد توزیع محبوبیت را مطابق با توزیع طبیعی Zipf مدل‌سازی کرده و دقت تشخیص را افزایش دهد[۱۲]. علیرغم تحقیقات موجود درباره حملات آلودگی حافظه نهان[۷] در شبکه‌های داده‌های نام‌گذاری‌شده(NDN) ، همچنان خلأهای اساسی در این حوزه وجود دارد که نیازمند بررسی بیشتر است. از جمله این موارد می‌توان به فقدان الگوریتم‌های مقاوم در برابر حملات چندمسیره، عدم وجود مدلی برای تنظیم حساسیت کَش بر اساس تغییرات رفتاری با استفاده از پارامتر β و توزیعZipf ، ابهام در نقش توزیع محبوبیت و تنوع ترافیک در آسیب‌پذیری کَش، کمبود مدل‌های سبک‌وزن قابل پیاده‌سازی و مقرون‌به‌صرفه برای سناریوهای واقعی، و نبود ترکیب مؤثر سیاست‌های جایگزینی کَش با مکانیزم‌های امنیتی اشاره کرد [۱۳].

با توجه به اینکه β تأثیر مستقیمی بر تعادل میان مقاومت در برابر آلودگی کَش و حفظ محتوای واقعی محبوب دارد، انتخاب مقدار مناسب آن در شرایط پویا و متغیر شبکه به‌صورت دستی یا استاتیک نه‌تنها دشوار بلکه ناکارآمد است. یکی از راهکارهای مؤثر برای تعیین مقدار بهینه پارامترβ ، بهره‌گیری از روش‌های بهینه‌سازی هوشمند و خودتطبیقی است. به طور کلی، تکنیک‌های بهینه سازی موجود را می توان به الگوریتم‌های قطعی[۸] و الگوریتم‌های فراابتکاری[۹](MAs) دسته بندی کرد [۱۴]. الگوریتم‌های قطعی توابع ریاضی خاصی هستند و به صورت مکانیکی و تکراری بدون هیچ گونه ماهیت تصادفی کار می کنند. در یک مسئله معین، یک روش قطعی همیشه خروجی یکسانی را برای یک ورودی خاص به دست می آورد. گرادیان[۱۰] نزولی و روش‌های نیوتن[۱۱]

[1] Information-Centric Networking
[2] Named Data Networking
[3] Cache Pollution Attack
[4] Least Recently Used
[5] Least Frequently Used
[6] Per-Face Popularity
[7] Cache Pollution Attacks
[8] Deterministic algorithms
[9] Metaheuristic algorithms
[10] Gradient descent
[11] Newton’s methods

دو نمونه کلاسیک از الگوریتم‌های قطعی هستند. از این‌رو، استفاده از الگوریتم‌های بهینه‌سازی مبتنی بر جست‌وجوی هوشمند مانند الگوریتم‌های تکاملی[1]، بهینه‌سازی ازدحام ذرات[2](PSO) ، الگوریتم ژنتیک(GA) یا روش‌هایی چون بهینه‌سازی بیزی[3]و روش های قطعی می‌تواند رویکردی کارآمد برای تنظیم خودکارβ باشد [۱۵, ۱۶]. هدف اصلی این روش‌ها، یافتن مقداری از β است که در یک بازه مشخص (مثلاً بین ۰.۰۱ تا ۱) بیشترین کارایی را در مقابله با حملات آلودگی کَش و بیشترین نرخ اصابت را در محیط شبکه ارائه دهد. این روش‌ها می‌توانند با ارزیابی مکرر عملکرد الگوریتم در پاسخ به سناریوهای مختلف حمله، مقادیر مختلف β را بررسی کرده و بهترین مقدار را در هر مرحله تعیین کنند. برخی از این الگوریتم‌ها حتی قادرند به صورت بلادرنگ و در زمان اجرای سیستم، با توجه به داده‌های ورودی جاری از شبکه (مانند میزان نوسان نرخ اصابت، نسبت خطا، تأخیر واکشی داده و ...) مقدار β را بازتنظیم کرده و خود را با شرایط جدید تطبیق دهند[۱۷].

روش ضرب‌کننده‌های لاگرانژ تفکیکی متناوب[4] (ADMM) برای مسائل بهینه‌سازی مقید و محدب با ساختارهای تجزیه‌پذیر مناسب است. استفاده از ADMM در تعیین مقدار بهینه پارامتر β در الگوریتمPFP-β ، مزایای قابل‌توجهی را در زمینه همگرایی پایدار، تحلیل‌پذیری و تطبیق‌پذیری فراهم می‌سازد [۱۸].

در مسئله بهینه‌سازی تعیین مقدار β باید به‌گونه‌ای تنظیم شود که بین دو هدف متضاد یعنی کاهش میزان آلودگی کَش و حفظ نرخ بالای اصابت محتوا توازن برقرار گردد. این امر را می‌توان به‌صورت یک مسئله بهینه‌سازی دوجزئی مدل‌سازی کرد که در آن تابع هدف از دو مؤلفه تشکیل می‌شود: یکی نمایانگر ضریب آلودگی حافظه نهان و دیگری نمایانگر نرخ اصابت کَش. با استفاده ازADMM ، می‌توان این دو مؤلفه را به‌صورت جداگانه مدل کرده، سپس به طور هم‌زمان و هماهنگ از طریق به‌روزرسانی‌های تکراری پارامتر β را در هر گام به‌گونه‌ای تنظیم کرد که تعادل بهینه حاصل شود. ازآنجاکه ساختار PFP-β به طور ذاتی بر پایه تجمیع و نرمال‌سازی محبوبیت محتوا از چندین درگاه ورودی - خروجی است، استفاده از ADMM این امکان را فراهم می‌سازد که این تجمیع در هر درگاه به‌صورت موازی انجام شده و در نهایت نتایج بهینه‌سازی از طریق یک گام تلفیق[5] یکپارچه شوند. در چنین حالتی، هر Face نقش یک عامل محلی را ایفا می‌کند که در بهینه‌سازی شرکت دارد و مقدار β نهایی از طریق مشارکت توزیعی تمامی بخش‌ها حاصل می‌شود.

هدف پژوهش حاضر، طراحی و ارزیابی یک مدل بهینه مبتنی بر بهینه‌ساز ADMM جهت بهبود مدیریت کَش و مقاوم‌سازی آن در برابر حملات آلودگی حافظه نهان در معماری NDN است. در این تحقیق، با استفاده از الگوریتم PFP-DA و افزودن پارامتر تنظیمیβ ، نسخه‌ای بهینه‌سازی‌شده از این الگوریتم ارائه خواهد شد که قادر است به طور مؤثر با حملات آلودگی حافظه نهان مقابله کرده و کارایی شبکه را بهبود بخشد. علاوه بر این، این تحقیق قصد دارد با ارائه یک راهکار سبک‌وزن و قابل‌پیاده‌سازی در شبیه‌سازndnSIM ، امکان استفاده عملی از مدل پیشنهادی را در سناریوهای واقعی فراهم کند؛ بنابراین نوآوری‌های این مقاله عبارت است از:

- ارائه نسخه پارامتریک PFP-β از الگوریتم PFP-DA باقابلیت تنظیم رفتار کَش بر اساس پارامترβ
- مدل‌سازی دقیق توزیع محبوبیت محتوا مطابق با توزیع Zipf با استفاده از پارامترβ
- استفاده از روش ADMM (تقسیم‌بندی ضرب آدم) برای بهینه‌سازی پارامتر β و تصمیم‌گیری کَش به‌صورت سریع و مقیاس‌پذیر

ساختار این مقاله به‌گونه‌ای طراحی شده است که در بخش دوم، پیشینه پژوهش مرتبط با حملات آلودگی کَش در معماری NDNبه‌صورت جامع بررسی می‌شود. در بخش سوم، جزئیات روش پیشنهادی شامل الگوریتم PFP-β و کاربرد روش ADMM برای بهینه‌سازی پارامتر β به طور کامل تشریح خواهد شد. بخش چهارم نیز به ارائه نتایج تجربی به‌دست‌آمده از شبیه‌سازی در محیط ndnSIM و تحلیل‌های کمی و کیفی مربوط به عملکرد الگوریتم اختصاص دارد. در نهایت، در بخش پنجم، نتیجه‌گیری نهایی مقاله ارائه شده و پیشنهادهایی برای پژوهش‌های آتی مطرح می‌گردد.

[1] Evolutionary Algorithms
[2] Particle Swarm Optimization (PSO)
[3] Bayesian Optimization
[4] Alternating Direction Method of Multipliers
[5] Consensus update

۲-پیشینه تحقیق

مطالعه‌ای توسط ماندااپاتی و راناورا[1] در این پژوهش نشان می‌دهند که سیاست‌های کَش سنتی مانندLRU ، به دلیل تمرکز صرف بر فراوانی استفاده، قادر به تمایز بین محتوای معتبر و درخواست‌های مخرب نیستند. آن‌ها با ترکیبی از شاخص‌های «محبوبیت محتوا» و «تازگی داده» روشی برای تشخیص زودهنگام و کاهش اثر حملات آلودگی کَش در محیط IoT-NDN ارائه می‌دهند که به نرخ شناسایی ۹۶٪ دست‌یافته است. بااین‌حال، این رویکرد موجب افزایش سربار محاسباتی در گره‌های مرزی شده و نیازمند نگهداری ساختارهای داده اضافی برای ردیابی هم‌زمان دو شاخص محبوبیت و تازگی است [۱۹].

یانگ[2] و همکاران در سال ۲۰۲۵ طرح دفاع واکنشی پیشگیرانه ترکیبی HPR-DS [3]را معرفی می‌کنند که با ترکیب ماژول‌های «پیشگیرانه» و «واکنشی» در لایه‌های میانی و مرزی شبکه و استفاده از تحلیل سری‌زمانی و خوشه‌بندی چندبعدی داده‌ها، خنثی‌سازی حملات[4] (IFA) را تشخیص می‌دهد. این روش با مدیریت منابع مبتنی بر پیش‌بینی، کیفیت خدمات کاربران (QoS) را در طول فعالیت حمله حفظ می‌کند. اما پیچیدگی بالای الگوریتم، نیاز به پردازش بلادرنگ و سربار زیاد در گره‌های مرزی از جمله معایب قابل‌توجه آن است [۲۰].

وانگ[5] و همکاران در سال ۲۰۲۴، مدل‌سازی رفتار انتقال محتوا به‌وسیله فرآیند مارکوف و ساخت ماتریس انتقال احتمال، تغییرات غیرعادی در تأخیر و مسیرهای داده را به‌عنوان نشانگر حملات آلودگی کَش شناسایی می‌کنند. برخلاف روش‌هایی که فرض ثبات توزیع محبوبیت محتوا را در نظر می‌گیرند، این الگوریتم توانایی ارائه دقت تشخیص بالا در شبکه‌های پویا و با تغییر مداوم رفتار کاربران را دارد. با این حال، محاسبه و نگهداری ماتریس‌های انتقال بزرگ و پیچیدگی تحلیلی بالا، مقیاس‌پذیری عملی آن را محدود می‌کند [۲۱].

فیشول حامدی و همکاران[6] در سال ۲۰۲۴ نشان می‌دهند که استفاده از یادگیری عمیق در محیط NDN می‌تواند ابزار مؤثری برای شناسایی و مهار حملات آلودگی کَش باشد. آن‌ها با توسعه الگوریتمی مبتنی بر شبکه‌های عصبی عمیق در شبیه‌ساز ndnSIM الگوهای رفتاری پنهان در داده‌ها را استخراج کرده و عملکرد بهتری نسبت به تکنیک‌های کلاسیک یا الگوریتم‌های احتمال محور ارائه می‌دهند. بااین‌وجود، پیچیدگی محاسباتی بسیار بالا، نیاز به داده‌های آموزشی گسترده و هزینه اجرایی سنگین در شبکه‌های مقیاس‌پذیر از مهم‌ترین محدودیت‌های این رویکرد محسوب می‌شود [۲۲].

جان باغ[7] و همکاران در سال ۲۰۲۳ الگوریتم PFP را معرفی کرده‌اند که با تحلیل «محبوبیت نرمال‌شده» محتوا بر اساس سهم هرFace ، از تمرکز مصنوعی محبوبیت توسط کاربران مشکوک جلوگیری می‌کند. نسخه PFP-DA با افزودن مکانیزم «کهنگی پویا» و نسخه PFP-β با به‌کارگیری پارامتر β برای شبیه‌سازی توزیعZipf ، دقت و مقاومت الگوریتم را در برابر حملات آلودگی کَش افزایش می‌دهند. این الگوریتم‌ها نتایج تجربی قابل‌توجهی در مقایسه با روش‌های رایج مانند LRU و LFU-DA ارائه کرده‌اند، اما پیاده‌سازی آن‌ها نیازمند نگهداری ساختارهای داده جداگانه برای هر درگاه و پردازش بلادرنگ تغییرات توزیع محبوبیت است که می‌تواند در گره‌هایی با منابع محدود به چالش بیانجامد [۱۲].

عبداهاک حیدوری[8] و همکاران در سال ۲۰۲۳ با معرفی الگوریتم Q-ICAN مبتنی بر یادگیری تقویتی، در هر روتر NDN عاملی قرار می‌دهند که با استفاده از شاخص‌هایی مانند نوسانات نرخ اصابت کَش (CHR) و فاصله زمانی بین درخواست‌ها، رفتار مشروع و مخرب را فرامی‌گیرد. این روش در شبیه‌سازی‌ها به‌دقت تشخیص ۹۵،۰۹٪، نرخ اصابت کَش ۹۴٪ و کاهش ۱۸٪ در زمان بازیابی داده‌ها رسیده است. بااین‌حال، نیاز به زمان طولانی برای آموزش، طراحی دقیق تابع پاداش و پیچیدگی محاسباتی بیشتر نسبت به الگوریتم‌های آماری ساده از معایب اصلی این رویکرد به شمار می‌آید [۲۳].

---

[1] Mandapati ،Ranaweera
[2] Yang
[3] Hybrid Proactive Reactive Defense Scheme
[4] Interest Flooding
[5] Wang
[6] Fishol Hamdi
[7] John Baugh
[8] Abdelhak Hidouri

با لیو و پنگ[1] در سال ۲۰۲۳ با ارائه الگوریتم BO-CatBoost که از بهینه‌سازی بیزی و ویژگی‌هایی مانند تعداد کَش از دست‌رفته و ورودی‌های PIT بهره می‌گیرد، توانسته‌اند هم‌زمان حملات آلودگی کَش و حملات پیشرفته سیل درخواست‌های هم‌دست‌شده[2] (I-CIFA)را تشخیص دهند. این روش، عملکرد پایداری در شناسایی دو نوع حمله هم‌زمان ارائه می‌دهد، اما نیاز به توان پردازشی بالا، فضای ذخیره‌سازی بیشتر و تنظیم دقیق پارامترهای یادگیری، کاربردپذیری آن را در محیط‌های واقعی محدود می‌کند [۲۴].

مان[3] و همکاران در سال ۲۰۲۱ الگوریتمی مبتنی بر درخت تصمیم‌گیری تقویت‌شده گرادیانی[4] (GBDT) برای تشخیص حملات آلودگی کَش در ICN ارائه کرده‌اند. در این روش، ویژگی‌های آماری و مسیریابی استخراج‌شده برای آموزش مدل استفاده می‌شوند تا رفتار غیرعادی در ترافیک شبکه تشخیص داده شود. نتایج آزمایش‌های مقایسه‌ای نشان داد که GBDT از نظر دقت تشخیص برتری قابل‌توجهی نسبت به روش‌های آستانه‌محور دارد. با این وجود، پیچیدگی محاسباتی بالا و نیاز به داده‌های آموزشی با کیفیت، پیاده‌سازی این روش را در گره‌های سبک‌وزن یا با منابع محدود دشوار می‌کند[۲۵].

یائو و همکاران [5] در (۲۰۲۰) روشی مبتنی بر پیش‌بینی محبوبیت با استفاده از مدل خاکستری پیشنهاد داده‌اند که با تحلیل روندهای تاریخی رفتار درخواست‌ها و برآورد سه شاخص اصلی—الگوی درخواست گذشته، محبوبیت لحظه‌ای و پیش‌بینی آینده محبوبیت—انحرافات غیرعادی در رفتار ترافیکی را شناسایی می‌کند. در صورت تشخیص اختلاف مکرر بین محبوبیت پیش‌بینی‌شده و واقعی، حمله شناسایی و سیاست دفاعی مناسب اعمال می‌شود. مزیت این رویکرد در تفکیک رفتار مشروع ناگهانی از حملات واقعی است، اما وابستگی به دقت مدل پیش‌بینی و نیاز به داده‌های تاریخی معتبر، از محدودیت‌های اجرایی آن محسوب می شود [۲۶].

ژائو[6] همکاران در سال ۲۰۲۰ در چارچوب V-NDN با تقسیم کَش هر گره به دو بخش «پرطرفدار» و «کم‌طرفدار»، محتوای ورودی را بر اساس سطح محبوبیت در محل مناسب ذخیره می‌کنند و در صورت نوسان شدید محبوبیت، محتوا را به لیست نظارتی حمله اضافه می‌کنند. نتایج شبیه‌سازی‌ها نشان می‌دهد که این مکانیزم در مواجهه با حملات فعال تا ۱۴٪ نرخ اصابت کَش را ارتقاء داده و تا ۳۰٪ تأخیر پاسخگویی را کاهش می‌دهد؛ اما مدیریت پیچیده چندبخشی کَش و نیاز به به‌روزرسانی مداوم اطلاعات محبوبیت در محیط‌های پویای VANET می‌تواند منابع گره را تحت فشار قرار دهد [۲۷].

بووانساریی[7] و همکاران در سال ۲۰۲۰ الگوریتمی مبتنی بر شبکه عصبی تابع پایه شعاعی[8] RBFNNارائه داده‌اند که با دریافت ویژگی‌های ذاتی مرتبط با محتوای کَش‌شده، نوع حمله را به سه دسته «سالم»، «تخریب موضعیت» و «محبوبیت کاذب» طبقه‌بندی می‌کند. نتایج آزمایش‌ها نشان داد که RBFNN در شناسایی نوع حمله دقت بالایی دارد و از نظر کارایی نسبت به برخی روش‌های مرسوم عملکرد بهتری ارائه می‌دهد. بااین‌حال، نیاز به تنظیم دقیق پارامترها، طراحی مؤثر تابع فعال‌سازی و مدیریت سربار پردازشی در گره‌های لبه با محدودیت منابع، از چالش‌های اصلی این رویکرد است[۲۸].

نایگون[9] و همکاران در سال ۲۰۱۹ از طریق مدل تشخیصی مبتنی بر آزمون نسبت درست‌نمایی تعمیم‌یافته و یک آشکارساز ترتیبی، حملات سیل درخواست زیرساختی (IFA) را در پیاده‌سازی‌های واقعی ترکیبی NDN و IP بادقت بالا و بدون تحمیل سربار زیاد شناسایی می‌کنند. این تحقیق نشان می‌دهد که حتی باوجود استفاده از بسته‌های NACK برای کنترل جریان درخواست، مسیرهایی برای اجرای مؤثر حملات باقی می‌ماند. باوجود قابلیت اجراپذیری بالای این روش در محیط‌های ترکیبی NDN/IP، نیاز به توسعه بیشتر برای پوشش تمامی حملات در شبکه‌های صرفاً NDN و بهبود واکنش بلادرنگ وجود دارد [۲۹].

---

[1] Liu and Peng
[2] Improved Collusive Interest Flooding Attacks
[3] Man
[4] Gradient boost decision tree
[5] Yao
[6] Zhou
[7] Buvanesvari
[8] Radial Basis Function Neural Network
[9] Nguyen

سبا[1] و همکاران در سال ۲۰۱۹ چهار سناریوی تهدید ناشی از گره‌های فرصت‌طلب تأمین‌کننده محتوای جعلی[2](FCPs) را در شبکه‌های موبایل فرصت‌طلب[3] (OMN) تعریف کرده و با روش‌های بلک‌لیست دائمی یا موقتی، ارتباط با گره‌های مشکوک را مسدود می‌کنند. شبیه‌سازی‌ها نشان داد که در حضور ۴۰٪FCP ، نرخ پاسخ‌دهی کاربران قانونی تا ۴۰٪ کاهش یافته و تأخیر تا ۱۷۶٪ افزایش پیدا می‌کند؛ اما با اعمال روش‌های پیشنهادی تأخیر تا ۳۶٪ کاهش و نرخ پاسخ موفق تا ۹٪ افزایش می‌یابد. هرچند رویکرد بلک‌لیست مزایای قابل‌توجهی دارد، اما در مواجهه با تهدیدات پیچیده‌تر فرصت‌طلب ناکافی به نظر می‌رسد[۳۰].
باغ[4] و همکاران در سال ۲۰۱۸ در پایان‌نامه دکترای خود سه الگوریتمPFP ، PFP-DAو PFP-β را معرفی و ارزیابی کرده‌اند. در الگوریتمPFP ، محبوبیت محتوا بر اساس سهم هر Face محاسبه و نرمال‌سازی می‌شود تا از تمرکز مصنوعی محبوبیت جلوگیری گردد. نسخه PFP-DA با افزودن مکانیزم کهنگی پویا به داده‌هایی که صرفاً تاریخی محبوب بوده اما اکنون کم‌استفاده هستند اجازه می‌دهد تدریجاً حذف شوند، و نسخه PFP-β با پارامتر β توزیع محبوبیت را مطابق با Zipf شبیه‌سازی می‌کند. نتایج تجربی نشان داده‌اند که PFP-DA و PFP-β نسبت به LRU و LFU-DA در هر دو شرایط عادی و مواجهه با حملات آلودگی کَش عملکرد برتری دارند. اما اجرای این الگوریتم‌ها مستلزم محاسبات بلادرنگ و نگهداری اطلاعات انتشار در هر Face است که در گره‌های با منابع محدود به معارضاتی از نظر مصرف حافظه و پردازش می‌انجامد [۳۱].
در بررسی روش‌های مقابله با حملات کَش در معماری NDN می‌توان به موارد زیر اشاره کرد: اولاً، روش‌های مبتنی بر یادگیری ماشین، یادگیری تقویتی و تحلیل رفتاری پیشرفته دقت تشخیص بالایی دارند و می‌توانند الگوهای پیچیده ترافیک و رفتارهای غیرعادی را در محیط‌های پویا به‌خوبی شناسایی کنند. ثانیاً، مدل‌سازی رفتار کاربران، خوشه‌بندی داده‌ها و تحلیل مسیرهای محتوا توانسته است در برابر حملات چندوجهی و جعل محبوبیت عملکرد مؤثری ارائه دهد، اما این رویکردها به منابع پردازشی بالا و نگهداری ساختارهای داده سنگین نیاز دارند که اجرای آن‌ها در گره‌های مرزی یا سبک‌وزن دشوار است. ثالثاً، الگوریتم‌های تطبیق‌پذیر، نظیر روش‌های پیش‌بینی محبوبیت و تحلیل توزیع Zipf یا یادگیری تقویتی، توانایی نسبی بالایی در مواجهه با تغییرات سریع ترافیکی و حملات لحظه‌ای دارند؛ بااین‌حال، آن‌ها معمولاً به داده‌های آموزشی دقیق و زمان آموزش قابل‌توجهی نیازمندند که در محیط‌های واقعی محدودیت ایجاد می‌کند. در مقابل، روش‌های سبک‌وزن با ساختار ساده و قابلیت اجرا به‌صورت بلادرنگ، مناسبت بیشتری برای کاربرد در شبکه‌های زنده، گره‌های لبه و محیط‌های IoT یا VANET دارند؛ هرچند دقت آن‌ها معمولاً کمتر از مدل‌های پیشرفته است. در نهایت، انتخاب بهینه بین این رویکردها بستگی به محدودیت‌های منابع، نیاز به واکنش سریع و اهمیت دقت تشخیص در هر محیط عملیاتی دارد و توازن میان دقت، تطبیق‌پذیری و سادگی پیاده‌سازی، اساس طراحی سامانه‌ای مقاوم در برابر حملات کَش خواهد بود.

جدول ۱: مقایسه‌ای روش‌های مقابله با حملات کَش درNDN
Table 1: comparison of methods to deal with cache attacks in NDN

سادگی پیاده‌سازی	تحلیل بلادرنگ	نیاز به آموزش	مقیاس‌پذیری	سربار کم	تطبیق‌پذیری	مقاومت در برابر حمله	دقت تشخیص	روش پیشنهادی	شماره مرجع
✓	✓	✗	✓	✗	✓	✓	✓	محبوبیت + تازگی	[۱۹]
✗	✓	✗	✗	✗	✓	✓	✓	HPR-DS	[۲۰]

[1] Saha
[2] Fake Content Providers
[3] Opportunistic Mobile Networks
[4] Baugh

[۳۰]	لیست سیاه (OMN)	⚠	⚠	⚠	⚠	⚠	⚠	⚠	✓
[۲۹]	مدل LRT واقعی(	✓	✓	✓	✓	✓	✓	✗	✓
[۲۸]	شبکهRBF	✗	⚠	✓	✓	⚠	✓	✓	✓
[۲۷]	تقسیم کَش-V) NDN)	⚠	⚠	⚠	✓	✓	✓	✗	✓
[۲۶]	مدل خاکستری	⚠	⚠	⚠	✓	⚠	✓	✓	✓
[۲۵]	GBDT	✗	✓	✗	✓	✓	✓	✓	✓
[۲۴]	BO-CatBoost	✗	✗	✗	✓	✓	✓	✓	✓
[۳۲]	ICAN	✓	✗	✓	✗	⚠	✓	✓	✓
[۲۳]	Q-ICAN	✗	✓	⚠	⚠	⚠	✓	✓	✓
[۱۲]	PFP	⚠	✗	✓	⚠	⚠	✓	✓	✓
[۲۲]	یادگیری عمیق	✗	✗	✓	✗	✓	✓	✓	✓
[۲۱]	مدل مارکوف	✗	✗	✓	✗	✓	✓	✓	✓

✓	✓	×	✓	✓	✓	✓	✓	PFP-DA / β	[۳۱]

۳-روش پیشنهادی

در این بخش، رویکردی نوین برای تنظیم تطبیقی پارامتر β در الگوریتم PFP-β-DA ارائه می‌شود. پارامتر β نقش کلیدی در کنترل حساسیت الگوریتم نسبت به رفتار تقاضا در شبکه داده‌های نام‌گذاری‌شده (NDN) دارد؛ به‌گونه‌ای که مقادیر کوچک‌تر βموجب افزایش نرخ اصابت کَش و کاهش آلودگی می‌شود، در حالی که مقادیر بزرگ‌تر β می‌تواند به ذخیره‌سازی داده‌های غیرمحبوب منجر شود. نسخه اصلاح‌شده PFP-β-DA با بهبود فرمول محبوبیت، سهم درخواست‌ها را بین چهره‌های مختلف محتوا متعادل می‌کند و بدین ترتیب، محتوایی که از چند مسیر معتبر درخواست شده باشد، در کَش حفظ می‌شود و داده‌هایی که صرفاً از یک مسیر خاص تقاضا شده‌اند—حتی اگر ظاهراً پرتقاضا باشند—به راحتی کَش نمی‌شوند. این رویکرد منجر به افزایش دقت، کاهش آلودگی و بهبود عملکرد کَش در مواجهه با حملات می‌گردد. فرمول (۱) برای رتبه‌بندی درخواست‌های هر چهره‌ی محتوا و تعیین محبوبیت کلی آن درخواست‌ها در این روش استفاده می‌شود :

$$P(C,\beta)= \sum_{i=1}^{n} \frac{1}{r_i^{\beta}} \qquad (۱)$$
$$-\infty \le \beta \le +\infty$$

این فرمول محبوبیت کلی درخواست محتوای C را با درنظرگرفتن درخواست‌های هر دسته از محتوا C و استفاده از رتبه‌ی (r_i) محتوای C در هر Face (i) محاسبه می‌کند. پارامتر (r_i) نشان‌دهنده رتبه محتوای درخواست شده C در فهرست رتبه‌بندی محبوبیت Face (i) است. در این بخش، با توجه به ناکارآمدی مقدار ثابت β در شرایط پویا و متغیر شبکهNDN ، یک رویکرد تطبیقی برای تنظیم خودکار β ارائه می‌شود. برای این منظور، ابتدا مسئله بهینه‌سازی β مدل‌سازی شده و سپس با استفاده از روشADMM ، راهکاری جهت تنظیم لحظه‌ای و تطبیقی این پارامتر پیشنهاد شده است. این راهکار منجر به افزایش انعطاف‌پذیری و دقت الگوریتم کَش در برابر تغییرات ترافیکی و حملات خواهد شد.

۳-۱-تعریف مسئله بهنیه سازی ADMM

در این بخش، مسئله بهینه‌سازی تنظیم مقدار β به‌عنوان یک مسئله بهینه‌سازی پارامتری مدل‌سازی خواهد شد. در این مدل، دو معیار اصلی شبکه‌های داده‌های نام‌گذاری شده(NDN) ، یعنی نرخ اصابت[1] و درصد آلودگی کَش[2]، به‌عنوان توابعی از β در نظر گرفته می‌شوند. مقدار β به‌گونه‌ای تنظیم خواهد شد که نرخ اصابت بیشینه و درصد آلودگی کمینه شود. مدل مسئله *به‌صورت* زیر در نظر گرفته می‌شود:

$$\min_{\beta}(-HitRate(\beta) + \lambda.\,\text{PollutionRate}(\beta) + \frac{\rho}{2}(\beta - z + u)^2) \qquad (۲)$$
$$\text{Subject to } 0< \beta<1$$

که در رابطه بالا βبه‌عنوان متغیر تصمیم[3] شناخته می‌شود و در بازه [0, 1] تعریف شده است.Hit Rate(β) به‌عنوان تابع نرخ اصابت کَش تعریف خواهد شد. Pollution Rate(β) به‌عنوان تابع درصد آلودگی کَش در نظر گرفته خواهد شد. λ به‌عنوان پارامتر وزن‌دهی برای کنترل اهمیت نسبی آلودگی در مقابل نرخ اصابت تعریف خواهد شد. پارامتر وزنی که اهمیت کاهش آلودگی کَش[4] را نسبت به افزایش نرخ اصابت کَش تعیین می‌کند. ρ>0 پارامتر جریمه در ADMM که سرعت و پایداری همگرایی الگوریتم را کنترل می‌کند. Z متغیر کمکی در ADMM که حل مسئله را به زیر مسئله‌های ساده‌تر تقسیم می‌کند. **u** متغیر دوگان (ضریب لاگرانژ) که هماهنگی بین به‌روزرسانی‌های β و z را تضمین می‌کند. در الگوریتم ADMM (الگوریتم ضرب

[1] Hit Rate
[2] Pollution Rate
[3] Decision Variable
[4] PollutionRate

و اضافه‌سازی)، هدف اصلی تقسیم مسئله پیچیده به زیر مسئله‌های ساده‌تر است تا فرایند بهینه‌سازی را تسهیل کند. در این الگوریتم، سه مرحله اصلی به‌روزرسانی وجود دارد که هر یک از آن‌ها نقش خاصی در فرایند بهینه‌سازی دارند.

در **مرحله اول**، به‌روزرسانیβ ، تلاش می‌شود بهترین مقدار β که منجر به بهبود نرخ اصابت و کاهش درصد آلودگی کَش می‌شود، پیدا شود. در این مرحله، هدف بهینه‌سازی معادله (۲-۴) است که β را به‌گونه‌ای تنظیم می‌کند که هر دو معیار HitRateو PollutionRate در کنار هم بهینه شوند.

در **مرحله دوم**، به‌روزرسانی متغیر کمکی z انجام می‌شود. این مرحله تضمین می‌کند که مقدار β همواره در بازه [۰, ۱] باقی بماند و از محدوده مجاز خارج نشود. برای این منظور، از پروجکَشن[1] استفاده می‌شود تا هرگونه تغییر در β به‌گونه‌ای باشد که همچنان در محدوده مطلوب قرار گیرد.

در **مرحله سوم**، متغیر دوگان u به‌روزرسانی می‌شود تا هماهنگی بیشتری بین به‌روزرسانی‌های β و z حاصل شود. به طور خاص، uنقش یک عامل هماهنگ‌کننده را ایفا می‌کند و تغییرات β و z را به طور مؤثر کنترل می‌کند تا الگوریتم به‌صورت همگرایانه عمل کند. در نهایت، با انجام تکرارهای مکرر و بررسی شرایط همگرایی) با استفاده ازε_1 وε_2 برای β وz ، الگوریتم به سمت بهینه‌ترین نقطه همگرا می‌شود. در این فرآیند، تنظیم تطبیقی β که توسط ADMM انجام می‌شود موجب می‌شود که به‌طور مستمر به بهترین عملکرد در شرایط متغیر شبکه دست یابد. این فرآیند بهینه‌سازی تطبیقی به‌طور خاص در شبکه‌های داده‌های نام‌گذاری‌شده (NDN) مفید است زیرا شرایط و نیازهای شبکه ممکن است تغییر کند و نیاز به تنظیم بهینه β برای رسیدن به عملکرد بهتر وجود داشته باشد. به صورت خلاصه:

- در هر تکرار، الگوریتم مسئله را به زیر مسئله‌های ساده‌تر تقسیم می‌کند.
- β-update تلاش می‌کند بهترین β را برای بهبود HitRate و PollutionRate پیدا کند.
- z-update تضمین می‌کند β در بازه [۰,۱] باقی بماند.
- u-update هماهنگی بین β و zرا افزایش می‌دهد.
- در نهایت، β به‌صورت تطبیقی تنظیم می‌شود تا بهترین عملکرد را در شرایط متغیر شبکه حفظ کند. شبه‌کد مربوط به این روش در شکل ۱ به‌طور دقیق توضیح داده خواهد شد.

Initialize:
- Set initial values for β, z, u, ρ, λ
- Set convergence thresholds $\boldsymbol{\varepsilon_1}$, $\boldsymbol{\varepsilon_2}$
- Set maximum number of iterations MaxIter
- Set initial values for HitRate(β) and PollutionRate(β)

For k = 0 to MaxIter:

1. Update β using the minimization formula:

$$\boldsymbol{\beta}^{(k+1)} \leftarrow \arg\min_{\boldsymbol{\beta}} \left(-\boldsymbol{HitRate}(\boldsymbol{\beta}) + \boldsymbol{\lambda}.\mathbf{PollutionRate}(\boldsymbol{\beta}) + \frac{\boldsymbol{\rho}}{\mathbf{2}} \left(\boldsymbol{\beta} - \boldsymbol{z}^{(k)} + \boldsymbol{u}^{(k)}\right)^{\mathbf{2}}\right)$$

Subjet to 0< β<1

2. Update z using the projection operator:

$$\mathbf{z}^{(k+1)} \leftarrow \mathbf{Projection}_{[0\ 1]}(\boldsymbol{\beta}^{(k+1)} + \boldsymbol{u}^{(k)})$$

3. Update dual variable u:

$$\mathbf{u}^{(k+1)} \leftarrow \mathbf{u}^{(k+1)} + (\boldsymbol{\beta}^{(k+1)} + \boldsymbol{z}^{(k+1)})$$

4. Check for convergence:

If

$$\left|\boldsymbol{\beta}^{(k+1)} - \boldsymbol{\beta}^{(k)}\right| < \boldsymbol{\varepsilon_1}\ \boldsymbol{and}\ \left|\mathbf{z}^{(k+1)} - \boldsymbol{z}^{(k)}\right| < \boldsymbol{\varepsilon_2}$$

Break (converged)

End for

Return $\boldsymbol{\beta}^{(k+1)}, \mathbf{z}^{(k+1)}, \boldsymbol{u}^{(k+1)}$

شکل ۱: الگوریتم ADMM در بهبود حافظه نهان

Figure 1: ADMM algorithm in cache memory improvement

[1] Projection

۳-۲- ADMM بهبودیافته

در این بخش، برای ارتقای عملکرد الگوریتم ADMM در بهینه‌سازی تطبیقی پارامترβ ، روشی برای تنظیم خودکار دو پارامتر کلیدی یعنی ρ و λ معرفی می‌شود. پارامتر ρ به‌عنوان عامل جریمه درADMM ، نقش مهمی در تسریع همگرایی و تضمین پایداری الگوریتم دارد. تنظیم نامناسب آن ممکن است منجر به واگرایی یا کندی شدید در همگرایی شود. ازاین‌رو، استفاده از روش‌های تطبیقی برای تنظیم دینامیک ρ در طول تکرارهای الگوریتم، به‌ویژه از طریق روش‌های خطی یا غیرخطی، پیشنهاد می‌شود تا فرایند بهینه‌سازی با سرعت و دقت بیشتر انجام گیرد.

در کنار آن، پارامتر λ مسئول ایجاد تعادل میان دو هدف متضاد یعنی افزایش نرخ اصابت کَش و کاهش آلودگی آن است. تنظیم دقیق λ در محیط‌های پویا ضروری است تا سیستم بتواند به طور هم‌زمان به هر دو هدف پاسخ دهد. در این راستا، روش‌های بهینه‌سازی تطبیقی برای تنظیم لحظه‌ای λ نیز پیشنهاد می‌شود، تا تأثیر آن در هر گام از الگوریتم بر عملکرد کلی مدل بهینه شود. پارامتر ρ در روش ADMM به‌عنوان پارامتر جریمه نقش بسیار مهمی در همگرایی الگوریتم دارد. برای تنظیم خودکار ρ در طول فرایند بهینه‌سازی، می‌توان از یک قانون تطبیقی استفاده کرد. یکی از روش‌های معمول برای این کار، استفاده از قانون خطی یا غیرخطی برای تنظیم ρ بر اساس میزان تغییرات در متغیرهای β و z است. به‌روزرسانی ρ به‌صورت تطبیقی به‌صورت رابطه زیر است:

$$\rho^{(k+1)} = \rho^{(k)} . (1 + \alpha . \frac{\left|\beta^{(k+1)} - \beta^{(k)}\right|}{\max\left(|z^{(k+1)} - z^{(k)}|, \varepsilon\right)}) \quad (۳)$$

که این رابطه $\rho^{(k)}$مقدار فعلی پارامتر جریمه است. $\rho^{(k+1)}$ مقدار جدید پارامتر جریمه پس از به‌روزرسانی است. αیک ضریب ثابت برای کنترل سرعت تغییر ρ است.$\left|\beta^{(k+1)} - \beta^{(k)}\right|$ تغییرات در متغیر β در تکرارهای مختلف است. $\left|z^{(k+1)} - z^{(k)}\right|$ تغییرات در متغیر z در تکرارهای مختلف است. این فرمول باعث می‌شود که پارامتر ρ به‌طور خودکار و بر اساس میزان تغییرات در β و z تنظیم شود. افزایش ρ در صورتی که تغییرات در β و z زیاد باشد، همگرایی را تسریع می‌کند، در حالی که کاهش ρ زمانی که تغییرات کم باشد، از انحرافات بزرگ جلوگیری می‌کند. ε یک مقدار کوچک بزرگتر از صفر است. مقدار ε معمولاً عددی بسیار کوچک است که به‌طور خودکار تنظیم می‌شود تا از آسیب به فرآیند همگرایی جلوگیری کند.

- **تنظیم پارامتر λ به‌صورت تطبیقی**

برای تنظیم خودکار پارامتر λ که تأثیر زیادی در تعادل میان نرخ اصابت و آلودگی کَش دارد، می‌توان از یک روش مشابه برای تنظیم آن استفاده کرد. در اینجا، پیشنهاد می‌شود که λ بر اساس تغییرات در نرخ اصابت و درصد آلودگی کَش به طور خودکار به‌روزرسانی شود. فرمول پیشنهادی برای به‌روزرسانی λ به‌صورت تطبیقی به‌صورت زیر است:

$$\lambda^{(k+1)} = \lambda^{(k)} . (1 + \beta_\lambda . \frac{\left|HitRate\left(\beta^{(k+1)}\right) - HitRate\left(\beta^{(k)}\right)\right|}{\max(\left|\text{PollutionRate}(\beta^{(k+1)}) - \text{PollutionRate}(\beta^{(k)})\right|, \varepsilon)}) \quad (۴)$$

که در آن $\lambda^{(k)}$ مقدار فعلی پارامتر λ است. $\lambda^{(k+1)}$مقدار جدید پارامتر λ پس از به‌روزرسانی است. β_λیک ضریب ثابت برای کنترل سرعت تغییر λ است. $\left|HitRate\left(\beta^{(k+1)}\right) - HitRate\left(\beta^{(k)}\right)\right|$تغییرات در نرخ اصابت است. $\max(\left|\text{PollutionRate}\left(\beta^{(k+1)}\right) - \text{PollutionRate}\left(\beta^{(k)}\right)\right|, \varepsilon)$ بیشترین تغییرات در درصد آلودگی کَش است. در اینجا، λ با توجه به تغییرات در نرخ اصابت و آلودگی کَش تنظیم می‌شود. اگر تفاوت‌های زیادی در نرخ اصابت و آلودگی کَش مشاهده شود، λ به‌طور تطبیقی افزایش می‌یابد تا تعادل بهتری بین این دو معیار برقرار شود.

استفاده از max در مخرج، هدف اصلی جلوگیری از تقسیم بر صفر و همچنین بهینه‌سازی فرایند به‌روزرسانی پارامتر λاست. دلیل استفاده از max در مخرج این است که در صورت نزدیک بودن تغییرات نرخ اصابت و آلودگی به صفر، این تغییرات معمولاً خیلی کوچک هستند و ممکن است باعث نوسانات زیاد در روند همگرایی شوند. دراین‌رابطه، استفاده از تضمین می‌کند که هیچ‌گاه تقسیم بر صفر صورت نخواهد گرفت. به‌عبارت‌دیگر، اگر تفاوت‌های $\text{PollutionRate}\left(\beta^{(k+1)}\right)$ و $\text{PollutionRate}\left(\beta^{(k)}\right)$ بسیار کوچک و نزدیک به صفر باشند، به‌جای تقسیم بر عدد صفر، مقدار بسیار کوچکی ε در مخرج قرار داده می‌شود که مانع از تقسیم بر صفر و حفظ پایداری الگوریتم می‌شود. این امر در نهایت باعث بهینه‌سازی به‌طور ایمن و جلوگیری از تغییرات

غیرمنطقی در مقدار λ می‌شود و همگرایی الگوریتم را بهبود می‌بخشد. به‌طور کلی، هدف این است که حتی زمانی که تفاوت‌ها به حداقل ممکن رسیده‌اند، الگوریتم به‌طور پیوسته و بدون مشکل ادامه یابد.

- **فرمول کلی به‌روزرسانی ADMM با پارامترهای تطبیقی ρ وλ**

در این تحقیق، برای بهینه‌سازی عملکرد الگوریتم ADMM در حل مسئله بهینه‌سازی β و دستیابی به تعادل میان نرخ اصابت و آلودگی کَش، از تنظیم تطبیقی پارامترهای الگوریتم استفاده شده است. تنظیم مناسب پارامترهای ρ و λ نقش اساسی در تسریع همگرایی و بهبود دقت الگوریتم ایفا می‌کند. با بهره‌گیری از این رویکرد تطبیقی، به طور مؤثر تغییرات پارامترها در طول فرایند بهینه‌سازی مدیریت می‌شود که موجب بهبود کیفیت نتایج و کاهش هزینه‌های محاسباتی می‌گردد. این رویکرد علاوه بر اطمینان از همگرایی سریع و پایدار الگوریتم، به مدل این امکان را می‌دهد که در محیط‌های دینامیک و متغیر شبکه‌ها، عملکرد مطلوب خود را حفظ کند. شبکه کد روش پیشنهادی در الگوریتم شکل ۲ ارائه شده است. در آغاز اجرای الگوریتم، مقادیر اولیه متغیرهای اصلی شامل β متغیر تصمیم برای بهینه‌سازی نرخ اصابت و کاهش آلودگی، zمتغیر کمکی برای ساده‌سازی و هماهنگی، u متغیر دوگان برای تضمین همگرایی، و پارامترهای ρ و λ به‌ترتیب برای کنترل جریمه و تعادل بین نرخ اصابت و آلودگی) تعیین می‌شوند. همچنین، آستانه‌هایی برای همگرایی مشخص می‌شود تا الگوریتم در صورت رسیدن به شرایط مطلوب متوقف گردد.

مرحله ۱ : به‌روزرسانی β

در این مرحله، متغیر β به‌گونه‌ای به‌روزرسانی می‌شود که بهترین نتیجه را در زمینه بهبود نرخ اصابت و کاهش آلودگی کَش به دست آورد. هدف این است که β به‌گونه‌ای تنظیم شود که هر دو معیار بهینه شوند. این مرحله به‌طورکلی به حل یک مسئله بهینه‌سازی می‌پردازد که نیاز به استفاده از روش‌های بهینه‌سازی دارد.

$$\beta^{(k+1)} \leftarrow \arg\min_{\beta}(-HitRate(\beta) + \lambda^{(k)}.\,\text{PollutionRate}(\beta) + \frac{\rho^{(k)}}{2}\,(\beta - z^{(k)} + u^{(k)})^2) \qquad (۵)$$

مرحله ۲ : به‌روزرسانی متغیر کمکی z

در این مرحله، مقدار متغیر کمکی z به‌روزرسانی می‌شود تا همیشه در بازه معین قرار داشته باشد (در اینجا بازه[0, 1]). برای این منظور، عملگر projection به‌کار می‌رود تا اطمینان حاصل شود که مقدار z از این بازه خارج نمی‌شود.

$$\mathrm{z}^{(k+1)} \leftarrow \text{Projection}_{[0\ 1]}(\beta^{(k+1)} + u^{(k)}) \qquad (٦)$$

مرحله ۳ : به‌روزرسانی متغیر دوگان u

در این مرحله، متغیر دوگان u به‌روزرسانی می‌شود. این به‌روزرسانی برای همسان‌سازی و هماهنگی بین β و z است. وقتی β و z به طور مستقل به‌روزرسانی می‌شوند، متغیر u اختلافات میان این دو را اصلاح‌کرده و به‌روزرسانی‌های بعدی را هماهنگ می‌کند.

$$\mathrm{u}^{(k+1)} \leftarrow \mathrm{u}^{(k+1)} + (\beta^{(k+1)} + z^{(k+1)}) \qquad (۷)$$

مرحله ۴: به‌روزرسانی پارامتر ρ به‌صورت تطبیقی

در این مرحله، پارامتر ρ که به‌عنوان پارامتر جریمه در الگوریتم استفاده می‌شود، به طور تطبیقی تغییر می‌کند. اگر تغییرات در βیا z زیاد باشد، ρافزایش می‌یابد تا همگرایی سریع‌تر شود. در صورت تغییرات کم، ρکاهش می‌یابد تا از تغییرات غیرضروری جلوگیری شود.

$$\rho^{(k+1)} = \rho^{(k)}\,.\,(1 + \alpha.\frac{\left|\beta^{(k+1)} - \beta^{(k)}\right|}{\max\,(|z^{(k+1)} - z^{(k)}|, \varepsilon)}) \qquad (۸)$$

مرحله ۵: به‌روزرسانی پارامتر λ به‌صورت تطبیقی

در این مرحله، پارامتر λ که مسئول تنظیم تعادل بین نرخ اصابت و آلودگی کَش است، به طور تطبیقی تنظیم می‌شود. اگر تغییرات در نرخ اصابت زیاد باشد، λافزایش می‌یابد تا اهمیت این معیار بیشتر شود. اگر تغییرات در آلودگی کَش بیشتر باشد، λبه‌گونه‌ای تنظیم می‌شود که تأثیر بیشتری در کاهش آلودگی داشته باشد.

$$\lambda^{(k+1)} = \lambda^{(k)}.\left(1 + \beta_\lambda \cdot \frac{\left|HitRate\left(\beta^{(k+1)}\right) - HitRate\left(\beta^{(k)}\right)\right|}{\max(|\,\mathrm{PollutionRate}(\beta^{(k+1)}) - \mathrm{PollutionRate}(\beta^{(k)})|, \varepsilon)}\right) \quad (۹)$$

مرحله ۶ : همگرایی و توقف الگوریتم

الگوریتم با بررسی تغییرات در β و z به دنبال همگرایی می‌گردد. اگر تغییرات در هر دو متغیر از حد معینی کوچک‌تر شود، الگوریتم متوقف می‌شود. این تضمین می‌کند که الگوریتم به حالت بهینه خود رسیده است.

$$if\ \left|\beta^{(k+1)} - \beta^{(k)}\right| < \varepsilon_1\ and\ \left|z^{(k+1)} - z^{(k)}\right| < \varepsilon_2\ then\ stop \quad (۱۰)$$

در نهایت، پس از انجام تمام مراحل و به‌روزرسانی متغیرها، مقدار نهایی β ، z، λ و ρ به دست می‌آید که این مقادیر برای بهینه‌سازی سیستم در شرایط دینامیک و متغیر شبکه به کار می‌روند. تنظیم تطبیقی این پارامترها باعث تسریع همگرایی، بهبود عملکرد و کاهش هزینه‌های محاسباتی در الگوریتم می‌شود.

Initialize:
- Set initial values for β, z, u, ρ, λ
- Set convergence thresholds $\boldsymbol{\varepsilon_1}$, $\boldsymbol{\varepsilon_2}$
- Set constants α, β_λ
- Set maximum number of iterations MaxIter
- Set initial values for HitRate(β) and PollutionRate(β)

For k = 0 to MaxIter:

1. Update β using the minimization formula:

$$\boldsymbol{\beta^{(k+1)} \leftarrow \arg\min_{\beta}(-HitRate(\beta) + \lambda^{(k)}.\mathrm{PollutionRate}(\beta) + \frac{\rho^{(k)}}{2}\,(\beta - z^{(k)} + u^{(k)})^2)}$$

Apply optimization method to minimize the objective function

2. Update z using the projection operator:

$$\boldsymbol{z^{(k+1)} \leftarrow \mathrm{Projection}_{[0\ 1]}(\beta^{(k+1)} + u^{(k)})}$$

Ensure that z stays within the range [0, 1]

3. Update dual variable u:

$$\boldsymbol{u^{(k+1)} \leftarrow u^{(k+1)} + (\beta^{(k+1)} + z^{(k+1)})}$$

Adjust dual variable to enforce the consistency between β and z

4. Update ρ adaptively:

$$\boldsymbol{\Delta\beta \leftarrow |\beta\text{^}(k+1) - \beta\text{^}(k)|\ \ \Delta\beta \leftarrow \left|\beta^{(k+1)} - \beta^{(k)}\right|}$$

$$\boldsymbol{\Delta z \leftarrow |z\text{^}(k+1) - z\text{^}(k)|\ \Delta z \leftarrow \left|z^{(k+1)} - z^{(k)}\right|}$$

$$\boldsymbol{\rho^{(k+1)} \leftarrow \rho^{(k)}.\left(1 + \alpha.\frac{\Delta\beta}{\max(\Delta z, \varepsilon)}\right)}$$

Adjust the penalty parameter ρ based on changes in β and z

5. Update λ adaptively:

$$\boldsymbol{\Delta\mathrm{HitRate} \leftarrow \left|HitRate\left(\beta^{(k+1)}\right) - HitRate\left(\beta^{(k)}\right)\right|}$$

$$\boldsymbol{\Delta\mathrm{PollutionRate} \leftarrow \max(\left|\mathrm{PollutionRate}\left(\beta^{(k+1)}\right) - \mathrm{PollutionRate}\left(\beta^{(k)}\right)\right|, \varepsilon)}$$

$$\boldsymbol{\lambda^{(k+1)} \leftarrow \lambda^{(k)}.\left(1 + \beta_\lambda \cdot \frac{\Delta\mathrm{HitRate}}{\Delta\mathrm{PollutionRate}}\right)}$$

Adjust λ based on the change in the hit rate and pollution rate

6. Check for convergence:

if

$$\boldsymbol{\left|\beta^{(k+1)} - \beta^{(k)}\right| < \varepsilon_1\ and\ \left|z^{(k+1)} - z^{(k)}\right| < \varepsilon_2}$$

Break (converged)

If both β and z have converged, stop the algorithm

End for

Return $\boldsymbol{\beta^{(k+1)}, z^{(k+1)}, \lambda^{(k+1)}, \rho^{(k+1)}}$

شکل ۲: الگوریتم شبه کد پیشنهادی ADMM بهبود یافته

Figure 2: The pseudo-code algorithm of the proposed improved ADMM

۴-نتایج

در این پژوهش، برای انجام شبیه‌سازی‌ها و تحلیل‌ها، از ترکیب دو ابزار مکمل استفاده شده است تا هم جنبه‌های تحلیلی و عددی و هم ارزیابی‌های شبکه‌ای به‌صورت دقیق پوشش داده شود. پیاده‌سازی الگوریتم پیشنهادی و تحلیل رفتار بهینه‌سازی

پارامتر β در محیط MATLAB 2020Rb انجام شده است؛ این محیط بادقت بالا و توان محاسباتی مناسب، امکان تحلیل همگرایی، تنظیم پارامترها و بررسی عددی مدل را فراهم کرده است. در ادامه، برای ارزیابی عملکرد الگوریتم در سناریوهای واقعی شبکه‌هایNDN ، از شبیه‌ساز تخصصیndnSIM ، مبتنی برNS-3 ، بهره گرفته شد. این ابزار امکان شبیه‌سازی دقیق تعامل گره‌ها، حافظه نهان و شرایط حمله را فراهم می‌کند. سخت‌افزار مورداستفاده شامل پردازنده Intel Core i7 با فرکانس ۲.۶۰ گیگاهرتز و ۸ گیگابایت رم تحت سیستم‌عامل Windows 10 بوده است. این ترکیب ابزاری، توازن مؤثری میان تحلیل عددی دقیق و ارزیابی عملیاتی الگوریتم در محیط شبکه‌ای ایجاد کرده است.

۴-۱-معیارهای ارزیابی

در این پژوهش، برای ارزیابی عملکرد روش پیشنهادی، دو معیار اصلی به کار گرفته شده‌اند که نقش کلیدی در تحلیل و مقایسه الگوریتم‌ها دارند. اولین معیار، نرخ اصابت حافظه نهان[1](CHR) ، نشان‌دهنده‌ی درصد درخواست‌های پاسخ داده شده مستقیماً از کَش است و معیار اساسی برای سنجش کارایی سیستم کَش محسوب می‌شود؛ چرا که افزایش این نرخ منجر به کاهش تأخیر در تحویل محتوا و بهبود تجربه کاربر می‌شود.

$$\mathrm{CHR}\% = \frac{\text{Number of cache hits}}{\text{Total number of content requests}} \times 100 \qquad (۱۱)$$

دراین‌رابطه Number of cache hits تعداد درخواست‌هایی که از حافظه نهان پاسخ داده شده‌اند (تعداد اصابت‌ها به کَش) و Total number of content requestsتعداد کل درخواست‌های محتوای کاربران (صرف‌نظر از اینکه از کَش پاسخ داده شده‌اند یا نه) است.

دومین معیار، درصد آلودگی حافظه نهان[2](CPR) ، شاخصی مهم در شناسایی و کاهش محتوای غیرمفید و حملات آلودگی کَش است و بیانگر توانایی الگوریتم در حفظ کیفیت داده‌های ذخیره‌شده است.

$$\mathrm{CPR}\% = \frac{\text{Number of polluted contents in the cache}}{\text{TTotal number of cached contents}} \times 100 \qquad (۱۲)$$

دراین‌رابطه Number of polluted contents in the cache تعداد داده‌های آلوده یا غیرمحبوب موجود در حافظه نهان و Total number of cached contents تعداد کل داده‌هایی که در حافظه نهان ذخیره شده‌اند.

۴-۲-توپولوژی ها و سناریوها

در این پژوهش، چهار نوع توپولوژی شبکه شامل سلسله‌مراتبی، ساده، پیشرفته و مش (شکل ۳) طراحی شده‌اند تا عملکرد الگوریتم پیشنهادی به‌ویژه الگوریتم ADMM و نسخه بهبود یافته آن ارزیابی شود. در توپولوژی سلسله‌مراتبی در شکل ۳ a، شبکه شامل سه لایه است: لایه اول که گره‌های تولیدکننده محتوا را در بر می‌گیرد، لایه دوم که گره‌های میانی با حافظه‌های نهان هستند، و لایه سوم که گره‌های مصرف‌کننده در آن قرار دارند. در این توپولوژی، سه سناریو طراحی شده است: سناریوی اول شبکه پایدار است که به بررسی تأثیر ظرفیت کَش در سه مقدار مختلف می‌پردازد. سناریوی دوم به بررسی تاب‌آوری الگوریتم در برابر حملات آلودگی کَش (CPA) با افزایش تعداد درخواست‌های مخرب پرداخته است. در نهایت، سناریوی سوم تغییرات بار ترافیکی را مورد بررسی قرار می‌دهد که در آن نرخ درخواست‌ها به‌طور دینامیک از ۵ به ۲۰ درخواست در ثانیه تغییر می‌کند.

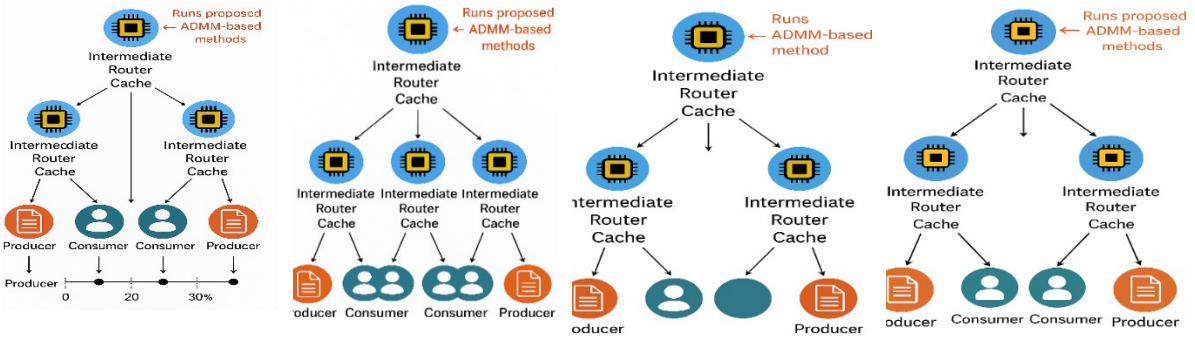
در توپولوژی ساده در شکل b، شبکه شامل سه‌لایه است: گره‌های تولیدکننده، مصرف‌کننده و میانه با حافظه نهان. در این توپولوژی نیز سه سناریو طراحی شده است: سناریوی شبکه پایدار که عملکرد الگوریتم در شرایط عادی بررسی می‌شود، حمله آلودگی کَش که شبیه‌سازی حملات با درخواست‌های جعلی را شامل می‌شود، و تغییرات بار ترافیکی که به بررسی توانایی الگوریتم در مقابله با تغییرات پویا در تعداد کاربران و نرخ درخواست‌ها می‌پردازد.

در توپولوژی پیشرفته در شکل c، ساختار پیچیده‌تری از سلسله‌مراتب با به‌روزرسانی پارامترهای الگوریتم ADMM در گره‌های میانی به‌صورت توزیع‌شده طراحی شده است. سه سناریو در این توپولوژی به شرح زیر است: در شبکه پایدار، عملکرد الگوریتم در شرایط بدون حمله ارزیابی می‌شود. در حمله آلودگی کَش(CPA) ، مقاومت الگوریتم در برابر حملات آلودگی کَش بررسی می‌شود و در تغییرات بار ترافیکی (پویا)، توانایی الگوریتم در پاسخ به تغییرات دینامیکی بار ترافیکی ارزیابی می‌شود.

---

[1] Cache Hit Rate
[2] Cache Pollution Rate

در توپولوژی مش در شکل d، گره‌ها به‌صورت غیرمتمرکز و با اتصالات چندگانه با یکدیگر در ارتباط هستند. در این توپولوژی، سه سناریو تعریف شده است :ازدحام موضعی و مسیرهای غیرتعادلی که به بررسی توانایی الگوریتم در مدیریت ترافیک غیر متعادل می‌پردازد، قطع برخی لینک‌ها که ارزیابی پایداری و تاب‌آوری الگوریتم در برابر خرابی‌های شبکه را شامل می‌شود، و حمله هدفمند توزیع‌شده که به بررسی مقاومت الگوریتم در برابر حملات توزیع‌شده آلودگی کَش می‌پردازد.



a) سلسله‌مراتبی b) ساده c) پیشرفته d) مش

شکل ۳: توپولوژی های بررسی شده

Figure 3: Topologies examined

این توپولوژی‌ها و سناریوها به طور جامع توانایی الگوریتم‌های پیشنهادی را در شرایط مختلف شبکه‌ای، از جمله حملات و تغییرات ترافیکی، ارزیابی می‌کنند. در ادامه جدول ۲ جامع برای توپولوژی‌ها و سناریوهای مختلف شبیه‌سازی‌شده در این پژوهش آورده شده است.

جدول ۲: بررسی جامع توپولوژی‌ها و سناریوهای مختلف شبیه‌سازی‌شده

Table 2: comprehensive review of different simulated topologies and scenarios

توپولوژی شبکه	سناریو	هدف ارزیابی	شرح
توپولوژی سلسله‌مراتبی (شکل a)	شبکه پایدار	ارزیابی حساسیت الگوریتم به ظرفیت حافظه کَش و بررسی نرخ اصابت کَش و آلودگی کَش	بررسی عملکرد الگوریتم در شرایط عادی و بدون حمله، با تنظیم ظرفیت کَش در سه مقدار مختلف (۵۰۰، ۱۰۰۰، ۲۰۰۰ محتوا)
	حمله آلودگی کَش (CPA)	ارزیابی مقاومت الگوریتم در برابر حملات آلودگی کَش و حفظ کیفیت سرویس	بررسی تاب‌آوری الگوریتم در برابر حملات آلودگی کَش با افزایش تدریجی تعداد درخواست‌های مخرب از ۵ به ۲۰ درخواست در ثانیه
	تغییرات بار ترافیکی (پویا)	سنجش تطبیق‌پذیری الگوریتم در شرایط متغیر شبکه و حفظ عملکرد بهینه کَش	بررسی توانایی الگوریتم در مدیریت تغییرات ناگهانی یا تدریجی در تعداد کاربران و نرخ درخواست‌ها (از ۵ به ۲۰ درخواست در ثانیه)
توپولوژی ساده (شکل b)	شبکه پایدار	ارزیابی نرخ اصابت کَش (CHR) و آلودگی کَش (CPR)در شرایط عادی و ثابت بدون حمله	بررسی عملکرد الگوریتم در شرایط پایدار و بدون حمله در شبکه با بار ترافیکی ثابت و نرخ درخواست کاربران مطابق با توزیعZipf
	حمله آلودگی کَش (CPA)	بررسی مقاومت الگوریتم در برابر حملات آلودگی کَش و حفظ نرخ اصابت کَش و کاهش آلودگی کَش	بررسی عملکرد الگوریتم در برابر درخواست‌های جعلی و غیرمحبوب ارسال شده توسط مهاجمان به حافظه‌های نهان
	تغییرات بار ترافیکی (پویا)	بررسی توانایی الگوریتم در تطبیق با تغییرات بار ترافیکی و تنظیم خودکار پارامترهایβ ، ρوλ	بررسی تغییرات ناگهانی یا تدریجی در تعداد کاربران و نرخ درخواست‌ها و ارزیابی توانایی الگوریتم در تطبیق خود با شرایط تغییرات بار
توپولوژی پیشرفته	شبکه پایدار	سنجش دقت الگوریتم در شرایط بدون حمله و حفظ نرخ اصابت کَش و کاهش آلودگی کَش	بررسی عملکرد الگوریتم در محیط بدون حمله با ساختار پیچیده‌تر و چندلایه (تولیدکننده، حافظه‌های نهان، مصرف‌کننده)

(شکل c)	حمله آلودگی کَش (CPA)	ارزیابی مقاومت الگوریتم در برابر حملات آلودگی کَش و حفظ کیفیت کَش و بهینه‌سازی پارامترها	بررسی تاب‌آوری الگوریتم در برابر حملات آلودگی کَش با درخواست‌های جعلی و داده‌های غیرمحبوب ارسال‌شده توسط مهاجمان
	تغییرات بار ترافیکی (پویا)	سنجش انطباق‌پذیری الگوریتم در برابر تغییرات بار ترافیکی و حفظ عملکرد بهینه در شرایط دینامیک شبکه	بررسی توانایی الگوریتم در مدیریت تغییرات پویا در تعداد کاربران و نرخ درخواست‌ها و ارزیابی قدرت الگوریتم در تطبیق خود با شرایط متغیر شبکه
توپولوژی مش (شکل d)	ازدحام موضعی و مسیرهای غیرتعادلی	ارزیابی قدرت الگوریتم در توزیع متوازن بار و حفظ تعادل کَش‌ها و کاهش تأخیر در تحویل محتوا	بررسی بار ترافیکی سنگین در بخش‌های خاص شبکه و ارزیابی توانایی الگوریتم در مدیریت ازدحام و توزیع بار به طور هوشمندانه میان مسیرهای جایگزین
	قطع برخی لینک‌ها	ارزیابی انعطاف الگوریتم در شرایط خرابی لینک‌ها و حفظ کیفیت سرویس و تحویل محتوا بدون اختلال	شبیه‌سازی خرابی لینک‌ها و بررسی عملکرد الگوریتم در شرایط قطع یا اختلال لینک‌ها و شناسایی مسیرهای جایگزین برای حفظ عملکرد کَش
	حمله توزیع‌شده آلودگی کَش	سنجش مقاومت الگوریتم در برابر حملات توزیع‌شده آلودگی کَش و توانایی آن در حفظ کیفیت کَش و بهینه‌سازی محتوا	بررسی عملکرد الگوریتم در برابر حملات هماهنگ و توزیع‌شده که به‌صورت هم‌زمان از مسیرهای مختلف درخواست‌های جعلی و غیرمحبوب به شبکه ارسال می‌شود

۴-۳-ارزیابی در توپولوژی سلسله مراتبی

در این پژوهش، عملکرد الگوریتم‌های ADMM و نسخه بهبودیافته آن (ADMM-Enhanced) در سه سناریو مختلف شبکه در توپولوژی سلسله‌مراتبی بررسی شد. در سناریو شبکه پایدار (جدول ۳)، الگوریتم‌های ADMM و ADMM-Enhanced نسبت به روش‌های کلاسیک مانند LFU-DA و LRU عملکرد بهتری داشتند، با نرخ اصابت کَش بالاتر و درصد آلودگی کَش پایین‌تر. در سناریوی حمله آلودگی کَش (جدول ۴)، این الگوریتم‌ها توانستند مقاومت بیشتری در برابر حملات نشان دهند و آلودگی کَش را کاهش داده و نرخ اصابت کَش را حفظ کنند، در حالی که روش‌های کلاسیک آسیب‌پذیرتر بودند. در سناریوی تغییرات بار ترافیکی (جدول ۵)، الگوریتم‌های ADMM و ADMM-Enhanced با توانایی تطبیق‌پذیری بالا، در شرایط پویا عملکرد مطلوبی داشتند و نرخ اصابت کَش را حفظ کرده و آلودگی کَش را کنترل کردند. به طور کلی، این الگوریتم‌ها در تمام سناریوها نسبت به روش‌های مرسوم عملکرد برتری نشان داده‌اند و به‌عنوان یک راهکار مؤثر برای مدیریت حافظه‌های نهان در شبکه‌های داده‌محور پیشنهاد می‌شوند.

جدول ۳: سناریو ۱ توپولوژی سلسله مراتبی: شبکه پایدار (بدون حمله)

Table 3: Hierarchical topology scenario 1: stable network (no attack)

الگوریتم	نرخ اصابت کَش (CHR)	درصد آلودگی کَش (CPR)	نرخ اصابت Top 100	نرخ اصابت Top 50
PFP-DA	21.00%	10.00%	68.00%	78.00%
LFU-DA	15.00%	20.00%	55.00%	65.00%
LRU	12.00%	25.00%	48.00%	58.00%
ADMM	23.00%	8.00%	70.00%	80.00%
ADMM-Enhanced	**25.00%**	**7.00%**	**72.00%**	**82.00%**

جدول ۴: سناریو ۲ توپولوژی سلسله مراتبی: شبکه تحت حمله آلودگی کَش**(CPA)**

Table 4: Hierarchical Topology Scenario 2: Network Under CPA Attack

الگوریتم	نرخ اصابت کَش (CHR)	درصد آلودگی کَش (CPR)	نرخ اصابت Top 100	نرخ اصابت Top 50
PFP-DA	19.02%	15.00%	62.80%	75.53%
LFU-DA	12.67%	36.00%	41.17%	50.00%
LRU	8.82%	36.00%	28.72%	34.64%
ADMM	21.00%	12.00%	65.00%	77.00%
ADMM-Enhanced	**23.00%**	**10.00%**	**68.00%**	**80.00%**

جدول ۵: سناریو ۳ توپولوژی سلسله مراتبی: تغییرات بار ترافیکی (پویا)

Table 5: scenario 3 of hierarchical topology: traffic load changes (dynamic)

الگوریتم	نرخ اصابت کَش (CHR)	درصد آلودگی کَش (CPR)	نرخ اصابت Top 100	نرخ اصابت Top 50
PFP-DA	18.00%	14.00%	60.00%	70.00%
LFU-DA	11.00%	28.00%	45.00%	52.00%
LRU	9.00%	30.00%	40.00%	48.00%

ADMM	20.00%	11.00%	63.00%	75.00%
ADMM-Enhanced	**22.00%**	**9.00%**	**65.00%**	**77.00%**

۴-۴-نتایج ارزیابی در سناریوهای توپولوژی ساده

در این تحقیق، عملکرد الگوریتم‌های ADMM و نسخه پیشرفته آن (ADMM-Enhanced) در سه سناریوی مختلف شبکه مورد ارزیابی قرار گرفت. در سناریوی شبکه پایدار (جدول ۶)، الگوریتم‌های پیشنهادی عملکرد قابل توجهی نسبت به روش‌های کلاسیک مانند LFU-DA و LRU داشتند. این الگوریتم‌ها توانستند نرخ اصابت کَش بالاتری داشته باشند و همزمان درصد آلودگی کَش را در سطح پایین‌تری حفظ کنند. در حالی که روش‌های دیگر مانند LFU-DA و LRU نرخ اصابت کَش کمتری داشتند و درصد آلودگی کَش در آن‌ها بالاتر بود. در سناریوی حمله آلودگی کَش (CPA) (جدول ۷)، الگوریتم‌های ADMM و ADMM-Enhancedمقاومت بهتری نشان دادند و توانستند با وجود حمله، آلودگی کَش را به میزان چشمگیری کاهش دهند و نرخ اصابت کَش را در سطح بالایی حفظ کنند. در مقابل، الگوریتم‌های کلاسیک مانند LFU-DA و LRU در برابر حملات آسیب‌پذیر بودند و با افزایش آلودگی کَش و کاهش نرخ اصابت مواجه شدند. در سناریوی تغییرات بار ترافیکی (پویا) (جدول ۸)، الگوریتم‌های ADMM و ADMM-Enhanced توانایی تطبیق خود با تغییرات پویا در بار ترافیکی را نشان دادند. برخلاف روش‌های سنتی که در شرایط تغییرات ناگهانی عملکرد ضعیفی داشتند، این الگوریتم‌ها توانستند نرخ اصابت کَش را حفظ کرده و آلودگی کَش را در سطح مطلوبی کنترل کنند. به‌طور کلی، نتایج نشان می‌دهند که الگوریتم‌های ADMM و ADMM-Enhanced به‌دلیل استفاده از تنظیم تطبیقی پارامترها، در تمامی سناریوها، از جمله شبکه‌های پایدار، شرایط تحت حمله و وضعیت‌های پویا، عملکرد بهتری نسبت به روش‌های مرسوم دارند.

جدول ۶: سناریو ۱ توپولوژی ساده: شبکه پایدار (بدون حمله)

Table 6: Scenario 1 Simple topology: stable network (no attack)

الگوریتم	(CHR) نرخ اصابت کَش	(CPR) درصد آلودگی کَش	Top 100 نرخ اصابت	Top 50 نرخ اصابت
PFP-DA	20.00%	11.00%	67.00%	77.00%
LFU-DA	13.00%	21.00%	53.00%	63.00%
LRU	10.00%	27.00%	47.00%	57.00%
ADMM	22.00%	9.00%	69.00%	79.00%
ADMM-Enhanced	**24.00%**	**8.00%**	**71.00%**	**81.00%**

جدول ۷: سناریو ۲ توپولوژی ساده: حمله آلودگی کَش**(CPA)**

Table 7: Scenario 2 of simple topology: CPA attack

الگوریتم	(CHR) نرخ اصابت کَش	(CPR) درصد آلودگی کَش	Top 100 نرخ اصابت	Top 50 نرخ اصابت
PFP-DA	18.50%	16.00%	61.00%	73.00%
LFU-DA	11.50%	35.00%	40.00%	49.00%
LRU	7.50%	37.00%	27.00%	33.00%
ADMM	20.00%	13.00%	64.00%	75.00%
ADMM-Enhanced	**22.00%**	**11.00%**	**67.00%**	**78.00%**

جدول ۸: سناریو ۳ توپولوژی ساده: تغییر بار ترافیکی (پویا)

Table 8: scenario 3 simple topology: traffic load change (dynamic)

الگوریتم	(CHR) نرخ اصابت کَش	(CPR) درصد آلودگی کَش	Top 100 نرخ اصابت	Top 50 نرخ اصابت
PFP-DA	17.00%	15.00%	58.00%	68.00%
LFU-DA	10.00%	29.00%	44.00%	50.00%
LRU	8.00%	31.00%	39.00%	46.00%
ADMM	19.00%	12.00%	62.00%	73.00%
ADMM-Enhanced	**21.00%**	**10.00%**	**65.00%**	**76.00%**

۴-۵-نتایج ارزیابی برای توپولوژی پیشرفته

الگوریتم‌های ADMM و نسخه پیشرفته آن (ADMM-Enhanced) در سه سناریو مختلف توپولوژی پیشرفته ارزیابی شدند. در سناریوی شبکه پایدار (جدول ۹)، الگوریتم‌های پیشنهادی عملکرد بهتری نسبت به روش‌های کلاسیک مانند LRU و-LFU DAداشتند. این الگوریتم‌ها توانستند نرخ اصابت کَش بالاتری را ثبت کنند و در عین حال، درصد آلودگی کَش را به‌طور قابل توجهی کاهش دهند. همچنین، در این سناریو، نرخ اصابت برای محتوای پرطرفدار Top 50 و Top 100 در الگوریتم‌های ADMM و نسخه پیشرفته آن بالاترین میزان را داشت. در سناریوی حمله آلودگی کَش (CPA) (جدول ۱۰)، الگوریتم‌های ADMM و

ADMM-Enhancedبه‌طور مؤثری آلودگی کَش را کاهش دادند و توانستند تعادل مناسبی بین نرخ اصابت و مقاومت در برابر آلودگی ایجاد کنند. در حالی که روش‌های کلاسیک نظیر LRU و LFU-DA با کاهش قابل توجه در نرخ اصابت و افزایش آلودگی کَش مواجه شدند. در سناریوی تغییرات بار ترافیکی (پویا) (جدول ۱۱)، الگوریتم‌های ADMM و ADMM-Enhanced عملکرد پایدار و تطبیق‌پذیری خوبی در شرایط متغیر و پویا از خود نشان دادند. این الگوریتم‌ها توانستند نرخ اصابت کَش را در سطح مطلوب حفظ کرده و آلودگی کَش را کنترل کنند. در مقابل، روش‌های مرجع در شرایط تغییرات بار ترافیکی ضعیف عمل کردند. به‌طور کلی، نتایج این سناریوها نشان می‌دهند که الگوریتم‌های ADMM و ADMM-Enhanced در تمامی شرایط از نظر نرخ اصابت کَش، مقاومت در برابر آلودگی و مدیریت محتواهای پربازدید عملکرد بهتری نسبت به روش‌های مرجع دارند و به‌عنوان گزینه‌ای ایده‌آل برای شبکه‌های NDN پیشنهاد می‌شوند.

جدول ۹: سناریو توپولوژی پیشرفته ۱: شبکه پایدار (بدون حمله)

Table 9: advanced topology scenario 1: stable network (no attack)

الگوریتم	(CHR) نرخ اصابت کَش	(CPR) درصد آلودگی کَش	Top 100 نرخ اصابت	Top 50 نرخ اصابت
PFP-DA	25.00%	8.00%	70.00%	80.00%
LFU-DA	18.00%	18.00%	60.00%	70.00%
LRU	15.00%	22.00%	52.00%	62.00%
ADMM	27.00%	6.00%	72.00%	82.00%
ADMM-Enhanced	**29.00%**	**5.00%**	**75.00%**	**85.00%**

جدول ۱۰: سناریو ۲ توپولوژی پیشرفته: شبکه تحت حمله آلودگی کَش**(CPA)**

Table 10: Advanced Topology Scenario 2: Network Under CPA Attack

الگوریتم	(CHR) نرخ اصابت کَش	(CPR) درصد آلودگی کَش	Top 100 نرخ اصابت	Top 50 نرخ اصابت
PFP-DA	23.00%	12.00%	65.00%	77.00%
LFU-DA	16.00%	30.00%	50.00%	60.00%
LRU	11.00%	32.00%	40.00%	48.00%
ADMM	25.00%	10.00%	68.00%	80.00%
ADMM-Enhanced	**27.00%**	**8.00%**	**70.00%**	**82.00%**

جدول ۱۱: سناریو ۳ توپولوژی پیشرفته: تغییرات بار ترافیکی (پویا)

Table 11: scenario 3 advanced topology: traffic load changes (dynamic)

الگوریتم	(CHR) نرخ اصابت کَش	(CPR) درصد آلودگی کَش	Top 100 نرخ اصابت	Top 50 نرخ اصابت
PFP-DA	21.00%	10.00%	63.00%	74.00%
LFU-DA	14.00%	24.00%	47.00%	56.00%
LRU	12.00%	26.00%	42.00%	52.00%
ADMM	24.00%	8.00%	66.00%	78.00%
ADMM-Enhanced	**26.00%**	**7.00%**	**68.00%**	**80.00%**

۴-۶-نتایج ارزیابی برای توپولوژی مش

در ارزیابی عملکرد الگوریتم‌های کَش در توپولوژی مش که در سه سناریو مهم بررسی شد، تفاوت‌های چشمگیری میان روش‌های کلاسیک و الگوریتم‌های مبتنی بر بهینه‌سازی توزیع‌شده مشاهده شد. در سناریوی شبکه پایدار (جدول ۱۲)، الگوریتم‌های کلاسیک با محدودیت‌هایی در حفظ تعادل بین نرخ پاسخ‌دهی محتوا و کیفیت کَش مواجه شدند، در حالی که الگوریتم‌های مبتنی بر ADMM با ساختار تطبیقی خود، توانستند کارایی بالاتری را از نظر ذخیره‌سازی محتوای پرطرفدار و کاهش داده‌های کم‌ارزش نشان دهند. در سناریوی حمله آلودگی کَش (جدول ۱۳)، تفاوت عملکرد میان روش‌های بهینه‌سازی‌شده و کلاسیک بیشتر شد. روش‌های سنتی که از فیلترهای ساده یا فرکانس دسترسی استفاده می‌کردند، در برابر محتوای آلوده آسیب‌پذیرتر بودند، در حالی که الگوریتم‌هایADMM ، به‌ویژه نسخه پیشرفته آن، توانستند با استفاده از کنترل تطبیقی و ارزیابی دقیق‌تر ارزش محتوا، اثر حملات را کاهش دهند و کیفیت داده‌های ذخیره‌شده را حفظ کنند. در سناریوی تغییرات بار ترافیکی (پویا) (جدول ۱۴)، الگوریتم‌های ADMM با انعطاف‌پذیری بالا در تنظیم ساختار کَش و تصمیم‌گیری توزیع‌شده، توانستند در شرایط متغیر بار، عملکرد مطلوبی از خود نشان دهند و کیفیت کَش را حفظ کنند، در حالی که روش‌های غیرهوشمند دچار افت کارایی شدند. این نتایج نشان می‌دهند که الگوریتم‌های پیشرفته مانندADMM ، به‌ویژه نسخه پیشرفته آن، در تمامی شرایط از جمله شبکه‌های پایدار، حملات آلودگی کَش و محیط‌های پویا، عملکرد برتری نسبت به الگوریتم‌های

کلاسیک دارند. استفاده از این الگوریتم‌ها در توپولوژی‌های پیچیده مانند مش، نه تنها مزایای فنی بلکه مزایای عملیاتی قابل توجهی برای شبکه‌های داده‌محور فراهم می‌آورد.

جدول ۱۲: سناریو ۱ توپولوژی مش : شبکه پایدار (بدون حمله)
Table 12: Mesh topology scenario 1: stable network (no attack)

الگوریتم	(CHR) نرخ اصابت کَش	(CPR) درصد آلودگی کَش	Top 100 نرخ اصابت	Top 50 نرخ اصابت
PFP-DA	20.00%	11.00%	67.00%	77.00%
LFU-DA	13.00%	21.00%	53.00%	63.00%
LRU	10.00%	27.00%	47.00%	57.00%
ADMM	22.00%	9.00%	69.00%	79.00%
ADMM-Enhanced	**24.00%**	**8.00%**	**71.00%**	**81.00%**

جدول ۱۳: سناریو ۲ توپولوژی مش : حمله آلودگی کَش**(CPA)**
Table 13: Mesh Topology Scenario 2: CPA

الگوریتم	(CHR) نرخ اصابت کَش	(CPR) درصد آلودگی کَش	Top 100 نرخ اصابت	Top 50 نرخ اصابت
PFP-DA	18.50%	16.00%	61.00%	73.00%
LFU-DA	11.50%	35.00%	40.00%	49.00%
LRU	7.50%	37.00%	27.00%	33.00%
ADMM	20.00%	13.00%	64.00%	75.00%
ADMM-Enhanced	**22.00%**	**11.00%**	**67.00%**	**78.00%**

جدول ۱۴: سناریو ۳ توپولوژی مش :تغییر بار ترافیکی (پویا)
Table 14: scenario 3 mesh topology: traffic load change (dynamic)

الگوریتم	(CHR) نرخ اصابت کَش	(CPR) درصد آلودگی کَش	Top 100 نرخ اصابت	Top 50 نرخ اصابت
PFP-DA	17.00%	15.00%	58.00%	68.00%
LFU-DA	10.00%	29.00%	44.00%	50.00%
LRU	8.00%	31.00%	39.00%	46.00%
ADMM	19.00%	12.00%	62.00%	73.00%
ADMM-Enhanced	**21.00%**	**10.00%**	**65.00%**	**76.00%**

۴-۷-مقایسه با مقاله پایه

در مقاله [۱۲]، دو سناریو برای بررسی تأثیر پارامتر β در الگوریتم PFP-β-DA ارائه شده است. سناریوی اول شرایط بدون حمله را بررسی می‌کند، که نشان می‌دهد کاهش β (به ویژه در محدوده ۰. ۵) باعث بهبود نرخ اصابت کَش و کاهش آلودگی کَش می‌شود. سناریوی دوم مربوط به شرایط حمله است که با افزایش تعداد مهاجمین، عملکرد PFP-β-DA کاهش می‌یابد، به‌ویژه زمانی که تعداد مهاجمین بیشتر از ۱۵٪ شود. در این تحقیق، الگوریتم پیشنهادی مبتنی بر ADMM با تنظیم خودکار پارامترهای ρو λ پیش‌بینی می‌شود که عملکرد بهتری نسبت به PFP-β-DA در شرایط پایدار و در مواجهه با حملات داشته باشد. این مقایسه به ارزیابی جامع‌تر روش پیشنهادی کمک می‌کند.

در جدول (۱۵) نتایج نشان می‌دهد که در شرایط پایدار بدون حمله، الگوریتم‌های ADMM و ADMM-Enhanced عملکرد بهتری نسبت به PFP-DA و PFP-β-DA دارند. PFP-β-DA با β بهینه (۰. ۵) توانسته آلودگی کَش را به صفر برساند، اما ADMM با بهینه‌سازی تطبیقی عملکرد بهتری در نرخ اصابت و کاهش آلودگی ثبت کرده است. نسخه پیشرفته ADMM (ADMM-Enhanced) نیز با تنظیم بهینه پارامترهای ρ و λ عملکرد بهتری در نرخ اصابت Top 100 و Top 50 داشت.

در سناریو دوم، جدول (۱۶)، که حمله با بار مهاجم ۵٪ بررسی شد، ADMMو ADMM-Enhanced به دلیل استفاده از تنظیم تطبیقی β توانستند نرخ اصابت کَش و نرخ اصابت برای محتوای پرطرفدار را نسبت به PFP-β-DA بهبود دهند و در برابر حملات مقاوم‌تر عمل کنند.

در سناریو سوم، جدول (۱۷)، که حمله با بار مهاجم ٪۳۰ بود، PFP-β-DA در برابر حملات سنگین عملکرد ضعیف‌تری نشان داد، زیرا β ایستا تغییرات لحظه‌ای را تطبیق نمی‌دهد. در مقابل، ADMM و ADMM-Enhanced با ساختار تطبیقی خود توانستند عملکرد بهتری در کاهش آلودگی و افزایش نرخ اصابت کَش داشته باشند. ADMM-Enhanced به ویژه توانست درصد آلودگی را به ٪۱۸ و نرخ اصابت Top 50 را به ٪۷۳ برساند، که نشان‌دهنده انعطاف‌پذیری بالای آن در شرایط حملات شدید است. این تحلیل‌ها نشان می‌دهند که الگوریتم‌های ADMM و ADMM-Enhanced در شرایط مختلف عملکرد بهتری نسبت

به PFP-β-DA دارند، به‌ویژه با استفاده از β تطبیقی که در مواجهه با حملات سنگین و شرایط متغیر شبکه به‌طور مؤثرتر عمل می‌کند.

جدول ۱۵: مقایسه عملکرد الگوریتم‌ها در سناریو ۱ (بدون حمله)
Table 15: comparing the performance of algorithms in scenario 1 (no attack)

الگوریتم	نرخ اصابت کَش (CHR)	درصد آلودگی کَش (CPR)	نرخ اصابت Top 100	نرخ اصابت Top 50
PFP-DA	21.3%	10.3%	68.2%	75.5%
PFP-β-DA (β=0.05)	25.0%	0.0%	93.1%	98.8%
ADMM	28.0%	0.0%	95.0%	99.0%
ADMM-Enhanced	**30.0%**	**0.0%**	**96.5%**	**99.5%**

جدول ۱۶: مقایسه عملکرد الگوریتم‌ها در سناریو ۲ (حمله با تعداد مهاجمین متنوع) با ۵٪ بار مهاجم
Table 16: comparing the performance of algorithms in scenario 2 (attack with a diverse number of attackers) with 5% attacker load

الگوریتم	نرخ اصابت کَش (CHR)	درصد آلودگی کَش (CPR)	نرخ اصابت Top 100	نرخ اصابت Top 50
PFP-DA	19.0%	10.0%	65.0%	75.0%
PFP-β-DA (β=0.05)	23.0%	0.0%	70.0%	80.0%
ADMM	26.0%	0.0%	74.0%	84.0%
ADMM-Enhanced	**28.0%**	**0.0%**	**77.0%**	**87.0%**

جدول ۱۷: مقایسه عملکرد الگوریتم‌ها در سناریو ۲ (حمله با تعداد مهاجمین متنوع) با ۳۰٪ بار مهاجم
Table 17: comparing the performance of algorithms in scenario 2 (attack with a diverse number of attackers) with 30% attacker load

الگوریتم	نرخ اصابت کَش (CHR)	درصد آلودگی کَش (CPR)	نرخ اصابت Top 100	نرخ اصابت Top 50
PFP-DA	10.0%	30.0%	50.0%	60.0%
PFP-β-DA (β=0.05)	12.0%	25.0%	55.0%	65.0%
ADMM	16.0%	20.0%	60.0%	70.0%
ADMM-Enhanced	**18.0%**	**18.0%**	**63.0%**	**73.0%**

۴-۸-مقایسه کیفی توپولوژی ها

در یک ارزیابی جامع در جدول ۱۸ از عملکرد الگوریتم‌ها در چهار توپولوژی مختلف و سناریوهای گوناگون، مشخص شد که روش‌های کلاسیک مانند LRU و LFU-DA در شرایط پویای شبکه و توپولوژی‌های پیچیده مانند مش عملکرد ضعیفی دارند. الگوریتم‌های آماری نظیر PFP-DA و PFP-β-DA در شرایط پایدار کارایی بهتری از خود نشان دادند، اما با افزایش بار حمله یا تغییرات شدید شبکه، دچار افت عملکرد شدند. در مقابل، الگوریتم‌های پیشنهادی ADMM و ADMM-Enhanced در تمام توپولوژی‌ها و سناریوها، از جمله حملات شدید، ازدحام یا تغییرات بار، عملکردی پایدار، تطبیق‌پذیر و مقاوم از خود نشان دادند. به‌ویژه ADMM-Enhanced با تنظیم دقیق پارامترهای تطبیقی، توانست کیفیت کَش و نرخ اصابت را حتی در شرایط استرس‌زای شبکه حفظ کند.

جدول ۱۸: مقایسه کیفی الگوریتم‌ها در تمامی توپولوژی‌ها و سناریوهای تعریف‌شده
Table 18: qualitative comparison of algorithms in all defined topologies and scenarios

توپولوژی / سناریو	الگوریتم	نرخ اصابت کَش(CHR)	مقاومت در برابر آلودگی(CPR)	حفظ محتوای محبوب	پایداری در بار متغیر	تطبیق‌پذیری ساختاری
ساده	LRU	ضعیف	ضعیف	متوسط	ضعیف	ضعیف
	LFU-DA	متوسط	ضعیف	متوسط	ضعیف	ضعیف
	PFP-DA	خوب	متوسط	خوب	متوسط	متوسط
	ADMM	خوب	خوب	خوب	خوب	خوب
	ADMM-Enhanced	عالی	عالی	عالی	عالی	بالا
سلسله‌مراتبی	LRU	ضعیف	بسیار ضعیف	ضعیف	ضعیف	ضعیف
	LFU-DA	ضعیف	ضعیف	ضعیف	ضعیف	ضعیف
	PFP-DA	متوسط	متوسط	متوسط	متوسط	متوسط
	ADMM	خوب	خوب	خوب	خوب	خوب

پیشرفته	ADMM-Enhanced	عالی	عالی	عالی	عالی	بالا
	LRU	ضعیف	ضعیف	ضعیف	ضعیف	ضعیف
	LFU-DA	ضعیف	ضعیف	ضعیف	ضعیف	ضعیف
	PFP-DA	متوسط	متوسط	متوسط	متوسط	متوسط
	ADMM	خوب	خوب	خوب	خوب	خوب
مش fault – tolerance	ADMM-Enhanced	عالی	عالی	عالی	عالی	بالا
	LRU	ضعیف	ضعیف	ضعیف	ضعیف	بسیار ضعیف
	LFU-DA	ضعیف	ضعیف	متوسط	ضعیف	ضعیف
	PFP-DA	متوسط	متوسط	خوب	متوسط	متوسط
	ADMM	خوب	خوب	خوب	خوب	خوب
مقاله پایه – بدون حمله(β=0.05)	ADMM-Enhanced	عالی	عالی	عالی	عالی	بالا
	PFP-β-DA	خوب	عالی	عالی	خوب	متوسط
	ADMM	عالی	عالی	عالی	عالی	بالا
مقاله پایه – حمله ۳۰٪ مهاجم	ADMM-Enhanced	عالی+	عالی+	عالی+	عالی+	بسیار بالا
	PFP-β-DA	متوسط	متوسط	متوسط	ضعیف	متوسط
	ADMM	خوب	خوب	خوب	خوب	خوب
	ADMM-Enhanced	عالی	عالی	عالی	عالی	بالا

۵-نتیجه گیری

در این پژوهش، باهدف مقابله هوشمندانه و تطبیق‌پذیر با چالش‌های مدیریت کَش در معماری شبکه‌های داده‌های نام‌گذاری شده(NDN) ، الگوریتمی مبتنی بر ADMM و نسخه بهبودیافته آن (ADMM-Enhanced) طراحی و ارزیابی شد. این الگوریتم با تنظیم پویا و توزیع‌شده پارامتر کلیدیβ ، و همچنین بهینه‌سازی هم‌زمان λ وρ ، توانست در سناریوهای متنوع شامل شبکه پایدار، حملات آلودگی کَش و تغییرات شدید بار، عملکردی برتر نسبت به روش‌های مرجع مانندLRU، LFU-DA و حتیPFP-β-DA از خود نشان دهد. نتایج شبیه‌سازی در توپولوژی‌های مختلف—از ساده تا مش—نشان داد که ADMM-Enhanced در حفظ نرخ بالای اصابت کَش و کاهش چشمگیر آلودگی، به‌ویژه در شرایط ناپایدار و مهاجم پذیر، بسیار موفق بوده و از نظر عددی و عملی به‌عنوان یک راهکار مقیاس‌پذیر و مقاوم برای مدیریت هوشمند کَش در نسل آینده شبکه‌های داده‌محور قابل‌اتکا است.

مراجع:

[1] F. A. Karim, A. H. M. Aman, R. Hassan, K. Nisar, and M. Uddin, “Named Data Networking: A Survey on Routing Strategies,” IEEE Access, vol. 10, pp. 90254-90270, 2022.

[2] X. Zhang, Y. Zhou, D. Wu, Q. Z. Sheng, S. Riaz, M. Hu, and L. Xiao, “A Survey on Privacy-Preserving Caching at Network Edge: Classification, Solutions, and Challenges,” ACM Computing Surveys, vol. 57, no. 5, pp. 1-38, 2025.

[3] P. Kar, L. Chen, W. Sheng, C. F. Kwong, and D. Chieng, “Advancing NDN security: Efficient identification of cache pollution attacks through rank comparison,” Internet of Things, vol. 26, pp. 101142, 2024.

[4] R. Alubady, M. Salman, and A. S. Mohamed, “A review of modern caching strategies in named data network: Overview, classification, and research directions,” Telecommunication Systems, vol. 84, no. 4, pp. 581-626, 2023.

[5] N. Sadat, and R. Dai, “A Survey of Quality-of-Service and Quality-of-Experience Provisioning in Information-Centric Networks,” Network, vol. 5, no. 2, pp. 10, 2025.

[6] A. H. Magsi, A. Ghulam, S. Memon, K. Javeed, M. Alhussein, and I. Rida, “A machine learning-based attack detection and prevention system in vehicular named data networking,” Comput. Mater. Contin, vol. 77, no. 2, pp. 1445-1465, 2023.

[7] A. Karami, and M. Guerrero-Zapata, “An anfis-based cache replacement method for mitigating cache pollution attacks in named data networking,” Computer Networks, vol. 80, pp. 51-65, 2015.

[8] N. U. Saqib, and S. Isnain, “A Survey on Mitigation of Cache Pollution Attacks in NDN,” Acta Technica Jaurinensis, 2025.
[9] Z. N. S. ALMUDAYNI, “Improving Energy Efficiency and Network Lifetime in IoT Systems: A Novel Theoretical Framework and Experimental Validation,” La Trobe, 2025.
[10] N. Kumar, and S. Srivastava, “IBPC: An Approach for Mitigation of Cache Pollution Attack in NDN using Interface-Based Popularity,” Arabian Journal for Science and Engineering, vol. 49, no. 3, pp. 3241-3251, 2024.
[11] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, E. Biersack, and E. Kirda, “Privacy risks in named data networking: What is the cost of performance?,” ACM SIGCOMM Computer Communication Review, vol. 42, no. 5, pp. 54-57, 2012.
[12] J. Baugh, and J. Guo, “Enhancing Cache Robustness in Information-Centric Networks: Per-Face Popularity Approaches,” Network, vol. 3, no. 4, pp. 502-521, 2023.
[13] S. S. Ullah, S. Hussain, I. Ali, H. Khattak, and S. Mastorakis, “Mitigating content poisoning attacks in named data networking: a survey of recent solutions, limitations, challenges and future research directions,” Artificial Intelligence Review, vol. 58, no. 2, pp. 42, 2024.
[14] U. Sadana, A. Chenreddy, E. Delage, A. Forel, E. Frejinger, and T. Vidal, “A survey of contextual optimization methods for decision-making under uncertainty,” European Journal of Operational Research, vol. 320, no. 2, pp. 271-289, 2025.
[15] R. Tourani, S. Misra, T. Mick, and G. Panwar, “Security, privacy, and access control in information-centric networking: A survey,” IEEE communications surveys & tutorials, vol. 20, no. 1, pp. 566-600, 2017.
[16] Y. Xiao, H. Cui, R. A. Khurma, and P. A. Castillo, “Artificial lemming algorithm: a novel bionic meta-heuristic technique for solving real-world engineering optimization problems,” Artificial Intelligence Review, vol. 58, no. 3, pp. 84, 2025.
[17] T. Semwal, S. Jain, A. Mohanta, and A. Jain, “A hybrid CNN-SVM model optimized with PSO for accurate and non-invasive brain tumor classification,” Neural Computing and Applications, pp. 1-30, 2025.
[18] D. Han, and X. Yuan, “A note on the alternating direction method of multipliers,” Journal of Optimization Theory and Applications, vol. 155, pp. 227-238, 2012.
[19] S. G. Mandapati, C. Ranaweera, and R. Doss, "Early Detection and Mitigation of Cache-Based Attacks in IoT-NDN," Deakin University, 2025.
[20] J. Yang, K. Ding, K. Xue, J. Han, D. S. L. Wei, Q. Sun, and J. Lu, “HPR-DS: A Hybrid Proactive Reactive Defense Scheme Against Interest Flooding Attack in Named Data Networking,” IEEE Transactions on Networking, pp. 1-16, 2025.
[21] H. Wang, D. Man, S. Han, H. Wang, and W. Yang, "Detection and Defense of Cache Pollution Attack Using State Transfer Matrix in Named Data Networks." pp. 545-556.
[22] M. M. F. Hamdi, Z. Chen, and M. Radenkovic, "Mitigating Cache Pollution Attack Using Deep Learning in Named Data Networking (NDN)." pp. 432-442.
[23] A. Hidouri, H. Touati, M. Hadded, N. Hajlaoui, P. Muhlethaler, and S. Bouzefrane, “Q-ICAN: A Q-learning based cache pollution attack mitigation approach for named data networking,” Computer Networks, vol. 235, pp. 109998, 2023/11/01/, 2023.
[24] L. Liu, and S. Peng, "Detection of A Novel Dual Attack in Named Data Networking." pp. 1-8.
[25] D. Man, Y. Mu, J. Guo, W. Yang, J. Lv, and W. Wang, “Cache pollution detection method based on GBDT in information-centric network,” Security and Communication Networks, vol. 2021, no. 1, pp. 6658066, 2021.
[26] L. Yao, Y. Zeng, X. Wang, A. Chen, and G. Wu, “Detection and defense of cache pollution based on popularity prediction in named data networking,” IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 6, pp. 2848-2860, 2020.
[27] J. Zhou, J. Luo, L. Deng, and J. Wang, "Cache pollution prevention mechanism based on cache partition in V-NDN." pp. 330-335.
[28] R. M. Buvanesvari, and K. Suresh Joseph, “RBFNN: A radial basis function neural network model for detecting and mitigating the cache pollution attacks in named data networking,” IET Networks, vol. 9, no. 5, pp. 255-261, 2020.
[29] T. Nguyen, H. L. Mai, R. Cogranne, G. Doyen, W. Mallouli, L. Nguyen, M. E. Aoun, E. M. D. Oca, and O. Festor, “Reliable Detection of Interest Flooding Attack in Real Deployment of Named Data Networking,” IEEE Transactions on Information Forensics and Security, vol. 14, no. 9, pp. 2470-2485, 2019.
[30] B. K. Saha, and S. Misra, “Mitigating NDN-based fake content dissemination in opportunistic mobile networks,” IEEE Transactions on Mobile Computing, vol. 19, no. 6, pp. 1375-1386, 2019.
[31] J. P. Baugh, “Enhancing Cache Robustness in Named Data Networks,” 2018.

[32] A. Hidouri, H. Touati, M. Hadded, N. Hajlaoui, and P. Muhlethaler, "A detection mechanism for cache pollution attack in named data network architecture." pp. 435-446.