

پژوهشی

مقایسه عملکرد الگوریتم های کش در شبکه های داده های نام گذاری شده با تمرکز بر کاهش حملات آلودگی و بهینه سازی پارامتر β

عاطفه واعظ شهرستانی^۱ | محمدرضا خیام باشی^۲ | فرامرز صافی^۳

چکیده:

مدیریت هوشمند حافظه نهان در شبکه های داده های نام گذاری شده (NDN)، یک مؤلفه حیاتی در بهینه سازی کارایی شبکه و ارتقای کیفیت تجربه کاربر است. تنظیم حساسیت کش بر اساس تغییرات رفتاری با استفاده از پارامتر β و توزیع Zipf در مدیریت حافظه نهان بسیار مهم است. با توجه به اینکه β تأثیر مستقیمی بر تعادل میان مقاومت در برابر آلودگی کش و حفظ محتوای واقعی محبوب دارد، تنظیم و تعیین این پارامتر نقشی بنیادین ایفا می کند. در این پژوهش، چارچوبی نوین مبتنی بر الگوریتم روش ضرایب لاگرانژ افزایشی جهت دار (ADMM) برای تنظیم توزیع شده و تطبیقی پارامتر β ارائه شده است. مدل پیشنهادی با ساختاری غیرمتمرکز، قابلیت اجرا در سطح گره های شبکه را داراست و از طریق به روزرسانی های هماهنگ متغیرهای اولیه و دوگانه، به همگرایی سریع و تنظیم دقیق β منتهی می شود. افزون بر آن، نسخه پیشرفته تری تحت عنوان ADMM-Enhanced توسعه یافته است که با بهره گیری از تنظیم خودکار پارامترهای ρ ، پایداری، دقت و تطبیق پذیری بیشتری نسبت به شرایط دینامیک شبکه حاصل می کند. برای ارزیابی کارایی این روش، چهار توپولوژی شبکه (ساده، سلسله مراتبی، پیشرفته و مش) در سناریوهای مختلف طراحی و شبیه سازی شد. نتایج شبیه سازی و مقایسه با الگوریتم های مرجع نشان دادند که الگوریتم ADMM-Enhanced در سناریوی پایدار نرخ اصابت کش 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، تطبیق پذیر و مقاوم از خود نشان داده است. همچنین در توپولوژی های پیچیده مانند مش، الگوریتم پیشنهادی توانست در برابر تهدیدات توزیع شده، ازدحام و قطع لینک، بهره وری و کیفیت کش را حفظ کرده و سازگاری بالایی با محیط های پویا از خود نشان دهد.

^۱ دانشکده مهندسی کامپیوتر - واحد نجف آباد - دانشگاه آزاد اسلامی - نجف آباد - ایران
atefehvaez@gmail.com

^۲ دانشکده مهندسی کامپیوتر - دانشگاه اصفهان - اصفهان - ایران
m.r.khayyambashi@eng.ui.ac.ir

^۳ مرکز تحقیقات کلان داده، - واحد نجف آباد - دانشگاه آزاد اسلامی - نجف آباد - ایران
fsafi@iaun.ac.ir

نویسنده مسئول

*محمدرضا خیام باشی،

M.r.khayyambashi@eng.ui.ac.ir

دانشکده مهندسی کامپیوتر - دانشگاه اصفهان -

اصفهان - ایران

تاریخ دریافت: ۱۴۰۴/۳/۱۵

تاریخ پذیرش: ۱۴۰۴/۴/۱۹

<https://doi.org/...>

کلید واژه ها: الگوریتم ADMM، ADMM-Enhanced، شبکه های داده های نام گذاری شده، حافظه نهان، نرخ اصابت کش، درصد آلودگی کش

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 میزبان-محور 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 محتوا-محور^۱ (ICN) مطرح شده‌اند که در آن‌ها تمرکز اصلی از «مکان محتوا» به «نام محتوا» منتقل می‌شود [۲، ۳]. یکی از مهم‌ترین نمونه‌های معماری‌های محتوا-محور، شبکه داده‌های نام‌گذاری شده^۲ (NDN) است. در این معماری، داده‌ها به صورت مستقل از مکان، نام‌گذاری شده و از طریق بسته‌های درخواستی و داده بین گره‌ها رد و بدل می‌شوند. یکی از مزایای کلیدی NDN، 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 در NDN، حملات آلودگی حافظه نهان^۳ (CPA) است. در این نوع حملات، مهاجم با ارسال درخواست‌های مکرر برای محتوای نامحسوب یا جعلی، موجب می‌شود داده‌های پرکاربرد از کش خارج شده و فضای کش با داده‌های کم‌ارزش پر شود. این فرآیند به کاهش نرخ اصابت کش، افت کارایی کلی شبکه و افزایش تأخیر پاسخ‌گویی می‌انجامد. بسیاری از سیاست‌های کش سنتی مانند کمترین استفاده اخیر LRU^۴ و کمترین استفاده شده به طور 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، PFP با 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 β (PFP- 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 نام‌گذاری شده (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 کش بر اساس تغییرات رفتاری با استفاده از پارامتر β و توزیع Zipf، ابهام در نقش توزیع محبوبیت و تنوع ترافیک در آسیب‌پذیری کش، کمبود مدل‌های سبک‌وزن قابل پیاده‌سازی و مقرون‌به‌صرفه برای سناریوهای واقعی، و نبود ترکیب مؤثر سیاست‌های جایگزینی کش با مکانیزم‌های امنیتی اشاره کرد [۱۳].

با توجه به اینکه β تأثیر مستقیمی بر تعادل میان مقاومت در برابر آلودگی کش و حفظ محتوای واقعی محبوب دارد، انتخاب مقدار مناسب آن در شرایط پویا و متغیر شبکه به صورت دستی یا استاتیک نه تنها دشوار بلکه ناکارآمد است. یکی از راهکارهای مؤثر برای تعیین مقدار بهینه پارامتر β ، بهره‌گیری از روش‌های بهینه‌سازی هوشمند و خودتطبیقی است. به طور کلی، تکنیک‌های بهینه‌سازی موجود را می‌توان 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 می‌کنند. در یک مسئله معین، یک روش قطعی همیشه خروجی یکسانی را برای یک ورودی خاص به دست می‌آورد. گرادیان^{۱۰} نزولی و روش‌های نیوتن^{۱۱}

¹ Information-Centric Networking² Named Data Networking³ Cache Pollution Attack⁴ Least Recently Used⁵ Least Frequently Used⁶ Per-Face Popularity⁷ Cache Pollution Attacks⁸ Deterministic algorithms⁹ Metaheuristic algorithms¹⁰ Gradient descent¹¹ Newton's methods

دو نمونه کلاسیک از الگوریتم های قطعی هستند. از این رو، استفاده از الگوریتم های بهینه سازی مبتنی بر جست و جوی هوشمند مانند الگوریتم های تکاملی^۱، بهینه سازی ازدحام ذرات (PSO)^۲، الگوریتم ژنتیک (GA) یا روش هایی چون بهینه سازی بیزی^۳ و روش های قطعی می تواند رویکردی کارآمد برای تنظیم خودکار β باشد [۱۵، ۱۶]. هدف اصلی این روش ها، یافتن مقداری از β است که در یک بازه مشخص (مثلاً بین ۰.۰۱ تا ۱) بیشترین کارایی را در مقابله با حملات آلودگی کش و بیشترین نرخ اصابت را در محیط شبکه ارائه دهد. این روش ها می توانند 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 الگوریتم ها حتی قادرند به صورت بلادرنگ و در زمان اجرای سیستم، با توجه به داده های ورودی جاری از شبکه (مانند میزان نوسان نرخ اصابت، نسبت خطا، تأخیر واکنشی داده و ...) مقدار β را بازتنظیم کرده و خود را با شرایط جدید تطبیق دهند [۱۷].

روش ضرب کننده های لاگرانژ تفکیکی متناوب^۴ (ADMM) برای مسائل بهینه سازی مقید و محدب با ساختارهای تجزیه پذیر مناسب است. استفاده از ADMM در تعیین مقدار بهینه پارامتر β در الگوریتم PFP- β ، مزایای قابل توجهی را در زمینه همگرایی پایدار، تحلیل پذیری و تطبیق پذیری فراهم می سازد [۱۸].

در مسئله بهینه سازی تعیین مقدار β باید به گونه ای تنظیم شود که بین دو هدف متضاد یعنی کاهش میزان آلودگی کش و حفظ نرخ بالای اصابت محتوا توازن برقرار گردد. این امر را می توان به صورت یک مسئله بهینه سازی دوجزئی مدل سازی کرد که در آن تابع هدف از دو مؤلفه تشکیل می شود: یکی نمایانگر ضریب آلودگی حافظه نهان و دیگری نمایانگر نرخ اصابت کش. با استفاده از ADMM، می توان این دو مؤلفه را به صورت جداگانه مدل کرده، سپس به طور همزمان و هماهنگ از طریق به روزرسانی های تکراری پارامتر β را در هر گام به گونه ای تنظیم کرد که تعادل بهینه حاصل شود. از آنجا که ساختار PFP- β به طور ذاتی بر پایه تجمیع و نرمال سازی محبوبیت محتوا از چندین درگاه ورودی - خروجی است، استفاده از ADMM این امکان را فراهم می سازد که این تجمیع در هر درگاه به صورت موازی انجام شده و در نهایت نتایج بهینه سازی از طریق یک گام تلفیق^۵ یکپارچه شوند. در چنین حالتی، هر Face نقش یک عامل محلی را ایفا می کند که در بهینه سازی شرکت دارد و مقدار β نهایی از طریق مشارکت توزیعی تمامی بخش ها حاصل می شود.

هدف پژوهش حاضر، طراحی و ارزیابی یک مدل بهینه مبتنی بر بهینه ساز ADMM جهت بهبود مدیریت کش و مقاوم سازی آن در برابر حملات آلودگی حافظه نهان در معماری NDN است. در این تحقیق، با استفاده از الگوریتم PFP-DA و افزودن پارامتر تنظیمی β ، نسخه ای بهینه سازی شده 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 سبک وزن و قابل پیاده سازی در شبیه ساز ndnSIM، امکان استفاده عملی از مدل پیشنهادی را در سناریوهای واقعی فراهم کند. بنابراین نوآوری های این مقاله عبارت است از:

- ارائه نسخه پارامتریک PFP- β از الگوریتم PFP-DA با قابلیت تنظیم رفتار کش بر اساس پارامتر β
 - مدل سازی دقیق توزیع محبوبیت محتوا مطابق با توزیع Zipf با استفاده از پارامتر β
 - استفاده از روش ADMM (تقسیم بندی ضرب آدم) برای بهینه سازی پارامتر β و تصمیم گیری کش به صورت سریع و مقیاس پذیر
- ساختار این مقاله به گونه ای طراحی شده است که در بخش دوم، پیشینه پژوهش مرتبط با حملات آلودگی کش در معماری NDN به صورت جامع بررسی می شود. در بخش سوم، جزئیات روش پیشنهادی شامل الگوریتم PFP- β و کاربرد روش ADMM برای بهینه سازی پارامتر β به طور کامل تشریح خواهد شد. بخش چهارم نیز به ارائه نتایج تجربی به دست آمده از شبیه سازی در محیط ndnSIM و تحلیل های کمی و کیفی مربوط به عملکرد الگوریتم اختصاص دارد. در نهایت، در بخش پنجم، نتیجه گیری نهایی مقاله ارائه شده و پیشنهادهایی برای پژوهش های آتی مطرح می گردد.

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 مانداپاتی و رانورا^۱ در این پژوهش نشان می‌دهند که سیاست‌های کش سنتی مانند LRU، 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 کش در محیط IoT-NDN ارائه می‌دهند که به نرخ شناسایی ۹۶٪ دست یافته است. با این حال، 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^۲ و همکاران در سال ۲۰۲۵ طرح دفاع واکنشی پیشگیرانه ترکیبی^۳ HPR-DS 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^۴ (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 قابل توجه آن است [۲۰].

وانگ^۵ و همکاران در سال ۲۰۲۴، مدل‌سازی رفتار انتقال محتوا به وسیله فرآیند مارکوف و ساخت ماتریس انتقال احتمال، تغییرات غیرعادی در تأخیر و مسیرهای داده را به عنوان نشانگر حملات آلودگی کش 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].

فیضول حامدی و همکاران^۶ در سال ۲۰۲۴ نشان می‌دهند که استفاده از یادگیری عمیق در محیط NDN می‌تواند ابزار موثری برای شناسایی و مهار حملات آلودگی کش باشد. آن‌ها با توسعه الگوریتمی مبتنی بر شبکه‌های عصبی عمیق در شبیه‌ساز IndnSIM الگوهای رفتاری پنهان در داده‌ها را استخراج کرده و عملکرد بهتری نسبت به تکنیک‌های کلاسیک یا الگوریتم‌های احتمال محور ارائه می‌دهند. با این وجود، 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^۷ و همکاران در سال ۲۰۲۳ الگوریتم PFP را معرفی کرده‌اند که با تحلیل «محبوبیت نرمال شده» محتوا بر اساس سهم هر Face، از تمرکز مصنوعی محبوبیت توسط کاربران مشکوک جلوگیری می‌کند. نسخه PFP-DA با افزودن مکانیزم «کهنگی پویا» و نسخه PFP- β با به کارگیری پارامتر β برای شبیه‌سازی توزیع Zipf، دقت و مقاومت الگوریتم را در برابر حملات آلودگی کش افزایش می‌دهند. این الگوریتم‌ها نتایج تجربی قابل توجهی 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^۸ 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 کش (CHR) و فاصله زمانی بین درخواست‌ها، رفتار مشروع و مخرب را فرا می‌گیرد. این روش در شبیه‌سازی‌ها به دقت تشخیص ۹۵٫۰۹٪، نرخ اصابت کش ۹۴٪ و کاهش ۱۸٪ در زمان بازبازی داده‌ها رسیده است. با این حال، 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^۱ در سال ۲۰۲۳ با ارائه الگوریتم BO-CatBoost که از بهینه‌سازی بیزی و ویژگی‌هایی مانند تعداد کش از دست‌رفته و ورودی‌های PIT بهره می‌گیرد، توانسته‌اند هم‌زمان حملات آلودگی کش و حملات پیشرفته سیل درخواست‌های هم‌دست‌شده^۲ (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^۳ و همکاران در سال ۲۰۲۱ الگوریتمی مبتنی بر درخت تصمیم‌گیری تقویت‌شده گرادیانی^۴ (GBDT) برای تشخیص حملات آلودگی کش 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^۵ 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 می‌شود [۲۶].

ژائو^۶ همکاران در سال ۲۰۲۰ در چارچوب V-NDN با تقسیم کش 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 کش را ارتقاء داده و تا ۳۰٪ تأخیر پاسخگویی را کاهش می‌دهد؛ اما مدیریت پیچیده چندبخشی کش و نیاز به به‌روزرسانی مداوم اطلاعات محبوبیت در محیط‌های پویای VANET می‌تواند منابع گره را تحت فشار قرار دهد [۲۷].

بووانساری^۷ و همکاران در سال ۲۰۲۰ الگوریتمی مبتنی بر شبکه عصبی تابع پایه شعاعی^۸ (RBFNN) ارائه داده‌اند که با دریافت ویژگی‌های ذاتی مرتبط با محتوای کش‌شده، نوع حمله را به سه دسته «سالم»، «تخریب موضعی» 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. با این حال، 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^۹ 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 با دقت بالا و بدون تحمیل سربار زیاد شناسایی می‌کنند. این تحقیق نشان می‌دهد که حتی با وجود استفاده از بسته‌های NACK برای کنترل جریان درخواست، مسیرهایی برای اجرای مؤثر حملات باقی می‌ماند. با وجود 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 Buvaesvari

⁸ Radial Basis Function Neural Network

⁹ Nguyen

سبا^۱ و هکاران در سال ۲۰۱۹ چهار سناریوی تهدید ناشی از گره‌های فرصت‌طلب تأمین‌کننده محتوای جعلی^۲ (FCPs) را در شبکه‌های موبایل فرصت‌طلب^۳ (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 قابل توجهی دارد، اما در مواجهه با تهدیدات پیچیده‌تر فرصت‌طلب ناکافی به نظر می‌رسد [۳۰]. باغ^۴ و همکاران در سال ۲۰۱۸ در پایان‌نامه دکترای خود سه الگوریتم PFP، PFP-DA و PFP-β را معرفی و ارزیابی کرده‌اند. در الگوریتم PFP، 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 کش 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 کش 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 به خوبی شناسایی کنند. ثانیاً، مدل‌سازی رفتار کاربران، خوشه‌بندی داده‌ها و تحلیل مسیرهای محتوا توانسته است در برابر حملات چندوجهی و جعل محبوبیت عملکرد موثری 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؛ با این حال، آن‌ها معمولاً به داده‌های آموزشی دقیق و زمان آموزش قابل توجهی نیازمندند که در محیط‌های واقعی محدودیت ایجاد می‌کند. در مقابل، روش‌های سبک‌وزن با ساختار ساده و قابلیت اجرا به صورت بلادرنگ، مناسب‌تری 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 کش خواهد بود.

جدول (۱) مقایسه‌ای روش‌های مقابله با حملات کش در NDN

Table (1) comparison of methods to deal with cache attacks in NDN

سادگی پیاده‌سازی	تحلیل بلادرنگ	نیاز به آموزش	مقیاس‌پذیری	مشارکت	تطبیق‌پذیری	مقاومت در برابر حمله	دقت تشخیص	روش پیشنهادی	شماره مرجع
▲	◀	×	▲	×	▲	▲	◀	محبوبیت + تاریخی	[۱۹]

¹ Saha² Fake Content Providers³ Opportunistic Mobile Networks⁴ Baugh

[۲۰]	HPR-DS	↖	↖	↖	×	×	×	↖	×
[۲۱]	مدل مارکوف	↖	↖	↖	×	×	×	↖	×
[۲۲]	یادگیری عمیق	↖	↖	↖	×	×	↖	↖	×
[۲۳]	PFP	↖	↖	⚠	⚠	↖	×	⚠	⚠
[۲۴]	Q-ICAN	↖	↖	↖	⚠	⚠	↖	↖	×
[۲۵]	ICAN	↖	⚠	×	↖	↖	×	⚠	↖
[۲۶]	BO-CatBoost	↖	↖	↖	×	×	↖	↖	×
[۲۷]	GBDT	↖	↖	↖	×	⚠	↖	↖	×
[۲۸]	مدل خاکستری	↖	⚠	↖	⚠	⚠	↖	↖	⚠
[۲۹]	تقسیم کش V (NDN)	↖	↖	↖	⚠	⚠	×	↖	⚠
[۳۰]	شبکه RBF	↖	↖	⚠	×	⚠	↖	↖	×
[۳۱]	مدل LRT واقعی	↖	↖	↖	↖	↖	×	↖	↖

$[x_0]$	پست سیاه (OMN)	▶	▶	◀	▶	▶	×	▶	◀
$[x_1]$	PFP-DA / β	◀	◀	◀	▶	◀	×	◀	▶

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 نام‌گذاری شده (NDN) دارد؛ به گونه‌ای که مقادیر کوچک‌تر β موجب افزایش نرخ اصابت کش 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 کش حفظ می‌شود و داده‌هایی که صرفاً از یک مسیر خاص تقاضا شده‌اند—حتی اگر ظاهراً پرتقاضا باشند—به راحتی کش نمی‌شوند. این رویکرد منجر به افزایش دقت، کاهش آلودگی و بهبود عملکرد کش در مواجهه با حملات می‌گردد. فرمول (۱) برای رتبه‌بندی درخواست‌های هر چهره‌ی محتوا و تعیین محبوبیت کلی آن درخواست‌ها در این روش استفاده می‌شود:

$$P(C, \beta) = \sum_{i=1}^n \frac{1}{r_i^\beta} \quad (1)$$

$$-\infty \leq \beta \leq +\infty$$

این فرمول محبوبیت کلی درخواست محتوای C را با در نظر گرفتن 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 شبکه NDN، یک رویکرد تطبیقی برای تنظیم خودکار β ارائه می‌شود. برای این منظور، ابتدا مسئله بهینه‌سازی β مدل‌سازی شده و سپس با استفاده از روش ADMM، راهکاری جهت تنظیم لحظه‌ای و تطبیقی این پارامتر پیشنهاد شده است. این راهکار منجر به افزایش انعطاف‌پذیری و دقت الگوریتم کش در برابر تغییرات ترافیکی و حملات خواهد شد.

۳-۱-تعریف مسئله بهینه سازی ADMM

در این بخش، مسئله بهینه‌سازی تنظیم مقدار β به عنوان یک مسئله بهینه‌سازی پارامتری مدل‌سازی خواهد شد. در این مدل، دو معیار اصلی شبکه‌های داده‌های نام‌گذاری شده (NDN)، یعنی نرخ اصابت^۱ و درصد آلودگی کش^۲، به عنوان تابعی از β در نظر گرفته می‌شوند. مقدار β به گونه‌ای تنظیم خواهد شد که نرخ اصابت بیشینه و درصد آلودگی کمینه شود. مدل مسئله به صورت زیر در نظر گرفته می‌شود:

$$\min_{\beta} (-HitRate(\beta) + \lambda \cdot PollutionRate(\beta) + \frac{\rho}{2} (\beta - z + u)^2) \quad (2)$$

$$\text{Subject to } 0 < \beta < 1$$

که در رابطه بالا β به عنوان متغیر تصمیم^۳ شناخته می‌شود و در بازه $(0, 1]$ تعریف شده است. $HitRate(\beta)$ به عنوان تابع نرخ اصابت کش تعریف خواهد شد. $PollutionRate(\beta)$ به عنوان تابع درصد آلودگی کش در نظر گرفته خواهد شد. λ به عنوان پارامتر وزنی برای کنترل اهمیت نسبی آلودگی در مقابل نرخ اصابت تعریف خواهد شد. پارامتر وزنی که اهمیت کاهش آلودگی

^۱ Hit Rate

^۲ Pollution Rate

^۳ Decision Variable

کش^۱ را نسبت به افزایش نرخ اصابت کش^۲ تعیین می کند. $\rho > 0$ پارامتر جریمه در ADMM که سرعت و پایداری همگرایی الگوریتم را کنترل می کند. Z متغیر کمکی در ADMM که حل مسئله را به زیرمسئله های ساده تر تقسیم می کند. u متغیر دوگان (ضریب لاگرانژ) که هماهنگی بین به روزرسانی های β و z را تضمین می کند. در الگوریتم ADMM (الگوریتم ضرب و اضافه سازی)، هدف اصلی تقسیم مسئله پیچیده به زیرمسئله های ساده تر است تا فرآیند بهینه سازی را تسهیل کند. در این الگوریتم، سه مرحله اصلی به روزرسانی وجود دارد که هر یک از آن ها نقش خاصی در فرآیند بهینه سازی دارند.

در **مرحله اول**، به روزرسانی β ، تلاش می شود بهترین مقدار β که منجر به بهبود نرخ اصابت و کاهش درصد آلودگی کش می شود، پیدا شود. در این مرحله، هدف بهینه سازی معادله (۲-۴) است که β را به گونه ای تنظیم می کند که هر دو معیار $PollutionRate$ و $HitRate$ در کنار هم بهینه شوند.

در **مرحله دوم**، به روزرسانی متغیر کمکی z انجام می شود. این مرحله تضمین می کند که مقدار β همواره در بازه $[0, 1]$ باقی بماند و از محدوده مجاز خارج نشود. برای این منظور، از پروجکشن^۳ استفاده می شود تا هرگونه تغییر در β به گونه ای باشد که همچنان در محدوده مطلوب قرار گیرد.

در **مرحله سوم**، متغیر دوگان u به روزرسانی می شود تا هماهنگی بیشتری بین به روزرسانی های β و z حاصل شود. به طور خاص، نقش یک عامل هماهنگ کننده را ایفا می کند و تغییرات β و z را به طور موثر کنترل می کند تا الگوریتم به صورت همگرایانه عمل کند. در نهایت، با انجام تکرارهای مکرر و بررسی شرایط همگرایی (با استفاده از ϵ_1 و ϵ_2 برای β و z ، الگوریتم به سمت بهینه ترین نقطه همگرا می شود. در این فرآیند، تنظیم تطبیقی β که توسط ADMM انجام می شود موجب می شود که به طور مستمر به بهترین عملکرد در شرایط متغیر شبکه دست یابد. این فرآیند بهینه سازی تطبیقی به طور خاص در شبکه های داده های نام گذاری شده (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 زیرمسئله های ساده تر تقسیم می کند.
- β -update تلاش می کند بهترین β را برای بهبود $HitRate$ و $PollutionRate$ پیدا کند.
- z -update تضمین می کند β در بازه $[0, 1]$ باقی بماند.
- u -update هماهنگی بین β و z را افزایش می دهد.
- در نهایت، β به صورت تطبیقی تنظیم می شود تا بهترین عملکرد را در شرایط متغیر شبکه حفظ کند. شبه کد مربوط به این روش در شکل ۱ به طور دقیق توضیح داده خواهد شد.

Initialize:

- Set initial values for $\beta, z, u, \rho, \lambda$
- Set convergence thresholds ϵ_1, ϵ_2
- Set maximum number of iterations $MaxIter$
- Set initial values for $HitRate(\beta)$ and $PollutionRate(\beta)$

For $k = 0$ to $MaxIter$:

1. Update β using the minimization formula:

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

Subject to $0 < \beta < 1$

2. Update z using the projection operator:

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

3. Update dual variable u :

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

4. Check for convergence:

¹ PollutionRate

² HitRate

³ Projection

If
 $|\beta^{(k+1)} - \beta^{(k)}| < \varepsilon_1 \text{ and } |z^{(k+1)} - z^{(k)}| < \varepsilon_2$
 Break (converged)
 End for
 Return $\beta^{(k+1)}, z^{(k+1)}, u^{(k+1)}$

شکل (۱) الگوریتم ADMM در بهبود حافظه نهان

Figure (1) ADMM algorithm in cache memory improvement

۳-۲-۲ ADMM بهبود یافته

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

در کنار آن، پارامتر λ مسئول ایجاد تعادل میان دو هدف متضاد یعنی افزایش نرخ اصابت کش و کاهش آلودگی آن است. تنظیم دقیق λ در محیط‌های پویا ضروری است تا سیستم بتواند به‌طور 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 فرآیند 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:

$$\rho^{(k+1)} = \rho^{(k)} \cdot \left(1 + \alpha \cdot \frac{|\beta^{(k+1)} - \beta^{(k)}|}{\max(|z^{(k+1)} - z^{(k)}|, \varepsilon)}\right) \quad (3)$$

که این رابطه $\rho^{(k)}$ مقدار فعلی پارامتر جریمه است. $\rho^{(k+1)}$ مقدار جدید پارامتر جریمه پس از به‌روزرسانی است. α یک ضریب ثابت برای کنترل سرعت تغییر ρ است. $|\beta^{(k+1)} - \beta^{(k)}|$ تغییرات در متغیر β در تکرارهای مختلف است. $|z^{(k+1)} - z^{(k)}|$ 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 کش 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 کش به‌طور خودکار به‌روزرسانی شود. فرمول پیشنهادی برای به‌روزرسانی λ به‌صورت تطبیقی به‌صورت زیر است:

$$\lambda^{(k+1)} = \lambda^{(k)} \cdot \left(1 + \beta_\lambda \cdot \frac{|HitRate(\beta^{(k+1)}) - HitRate(\beta^{(k)})|}{\max(|PollutionRate(\beta^{(k+1)}) - PollutionRate(\beta^{(k)})|, \varepsilon)}\right) \quad (4)$$

که در آن $\lambda^{(k)}$ مقدار فعلی پارامتر λ است. $\lambda^{(k+1)}$ مقدار جدید پارامتر λ پس از به‌روزرسانی است. β_λ یک ضریب ثابت برای کنترل سرعت تغییر λ است. $|HitRate(\beta^{(k+1)}) - HitRate(\beta^{(k)})|$ بیشترین تغییرات در نرخ اصابت است. $\max(|PollutionRate(\beta^{(k+1)}) - PollutionRate(\beta^{(k)})|, \varepsilon)$ با توجه به تغییرات در نرخ اصابت و آلودگی کش تنظیم می‌شود. اگر تفاوت‌های زیادی در نرخ اصابت و آلودگی کش 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 بهینه سازی فرآیند به روزرسانی پارامتر λ است. 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. در این رابطه، استفاده از تضمین می کند که هیچ گاه تقسیم بر صفر صورت نخواهد گرفت. به عبارت دیگر، اگر تفاوت های $PollutionRate(\beta^{(k+1)})$ و $PollutionRate(\beta^{(k)})$ بسیار کوچک و نزدیک به صفر باشند، به جای تقسیم بر عدد صفر، مقدار بسیار کوچکی ϵ در مخرج قرار داده می شود که مانع از تقسیم بر صفر و حفظ پایداری الگوریتم می شود. این امر در نهایت باعث بهینه سازی به طور ایمن و جلوگیری از تغییرات غیرمنطقی در مقدار λ می شود و همگرایی الگوریتم را بهبود می بخشد. به طور کلی، هدف این است که حتی زمانی که تفاوت ها به حداقل ممکن رسیده اند، الگوریتم به طور پیوسته و بدون مشکل ادامه یابد.

• فرمول کلی به روزرسانی ADMM با پارامترهای تطبیقی ρ و λ

در این تحقیق، برای بهینه سازی عملکرد الگوریتم ADMM در حل مسئله بهینه سازی β و دستیابی به تعادل میان نرخ اصابت و آلودگی کش، از تنظیم تطبیقی پارامترهای الگوریتم استفاده شده است. تنظیم مناسب پارامترهای ρ و λ نقش اساسی در تسریع همگرایی و بهبود دقت الگوریتم ایفا می کند. با بهره گیری از این رویکرد تطبیقی، به طور مؤثر تغییرات پارامترها در طول فرآیند بهینه سازی مدیریت می شود، که موجب بهبود کیفیت نتایج و کاهش هزینه های محاسباتی می گردد. این رویکرد علاوه بر اطمینان از همگرایی سریع و پایدار الگوریتم، به مدل این امکان را می دهد که در محیط های دینامیک و متغیر شبکه ها، 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 بهینه سازی نرخ اصابت و کاهش آلودگی، z متغیر کمکی برای ساده سازی و هماهنگی، u متغیر دوگان برای تضمین همگرایی، و پارامترهای ρ و λ به ترتیب برای کنترل جریمه و تعادل بین نرخ اصابت و آلودگی تعیین می شوند. همچنین، آستانه هایی برای همگرایی مشخص می شود تا الگوریتم در صورت رسیدن به شرایط مطلوب متوقف گردد.

مرحله ۱: به روزرسانی β

در این مرحله، متغیر β به گونه ای به روزرسانی می شود که بهترین نتیجه را در زمینه بهبود نرخ اصابت و کاهش آلودگی کش به دست آورد. هدف این است که β به گونه ای تنظیم شود که هر دو معیار بهینه شوند. این مرحله به طور کلی به حل یک مسئله بهینه سازی می پردازد که نیاز به استفاده از روش های بهینه سازی دارد.

$$\beta^{(k+1)} \leftarrow \arg \min_{\beta} (-HitRate(\beta) + \lambda^{(k)} \cdot PollutionRate(\beta) + \frac{\rho^{(k)}}{2} (\beta - z^{(k)} + u^{(k)})^2) \quad (5)$$

مرحله ۲: به روزرسانی متغیر کمکی z

در این مرحله، مقدار متغیر کمکی z به روزرسانی می شود تا همیشه در بازه معین قرار داشته باشد (در اینجا بازه $[0, 1]$). برای این منظور، عملگر projection به کار می رود تا اطمینان حاصل شود که مقدار z از این بازه خارج نمی شود.

$$z^{(k+1)} \leftarrow \text{Projection}_{[0, 1]}(\beta^{(k+1)} + u^{(k)}) \quad (6)$$

مرحله ۳: به روزرسانی متغیر دوگان u

در این مرحله، متغیر دوگان u به روزرسانی می شود. این به روزرسانی برای همسان سازی و هماهنگی بین β و z است. وقتی β و z به طور مستقل به روزرسانی می شوند، متغیر u اختلافات میان این دو را اصلاح کرده و به روزرسانی های بعدی را هماهنگ می کند.

$$u^{(k+1)} \leftarrow u^{(k+1)} + (\beta^{(k+1)} + z^{(k+1)}) \quad (7)$$

مرحله ۴: به روزرسانی پارامتر ρ به صورت تطبیقی

در این مرحله، پارامتر ρ که به عنوان پارامتر جریمه در الگوریتم استفاده می شود، به طور تطبیقی تغییر می کند. اگر تغییرات در β زیاد باشد، ρ افزایش می یابد تا همگرایی سریع تر شود. در صورت تغییرات کم، ρ کاهش می یابد تا از تغییرات غیرضروری جلوگیری شود.

$$\rho^{(k+1)} = \rho^{(k)} \cdot (1 + \alpha \cdot \frac{|\beta^{(k+1)} - \beta^{(k)}|}{\max(|z^{(k+1)} - z^{(k)}|, \varepsilon)}) \quad (8)$$

مرحله ۵: به‌روزرسانی پارامتر λ به‌صورت تطبیقی

در این مرحله، پارامتر λ که مسئول تنظیم تعادل بین نرخ اصابت و آلودگی کش است، به‌طور تطبیقی تنظیم می‌شود. اگر تغییرات در نرخ اصابت زیاد باشد، λ افزایش می‌یابد تا اهمیت این معیار بیشتر شود. اگر تغییرات در آلودگی کش بیشتر باشد، λ به‌گونه‌ای تنظیم می‌شود که تاثیر بیشتری در کاهش آلودگی داشته باشد.

$$\lambda^{(k+1)} = \lambda^{(k)} \cdot (1 + \beta_\lambda \cdot \frac{|HitRate(\beta^{(k+1)}) - HitRate(\beta^{(k)})|}{\max(|PollutionRate(\beta^{(k+1)}) - PollutionRate(\beta^{(k)})|, \varepsilon)}) \quad (9)$$

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.

$$if |\beta^{(k+1)} - \beta^{(k)}| < \varepsilon_1 \text{ and } |z^{(k+1)} - z^{(k)}| < \varepsilon_2 \text{ then stop} \quad (10)$$

در نهایت، پس از انجام تمام مراحل و به‌روزرسانی متغیرها، مقدار نهایی β ، z ، ρ و به‌دست 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 $\beta, z, u, \rho, \lambda$
- Set convergence thresholds $\varepsilon_1, \varepsilon_2$
- Set constants α, β_λ
- Set maximum number of iterations MaxIter
- Set initial values for $HitRate(\beta)$ and $PollutionRate(\beta)$

For $k = 0$ to MaxIter:

1. Update β using the minimization formula:

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

Apply optimization method to minimize the objective function

2. Update z using the projection operator:

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

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

3. Update dual variable u :

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

Adjust dual variable to enforce the consistency between β and z

4. Update ρ adaptively:

$$\Delta\beta \leftarrow |\beta^{(k+1)} - \beta^{(k)}| \quad \Delta\beta \leftarrow |\beta^{(k+1)} - \beta^{(k)}|$$

$$\Delta z \leftarrow |z^{(k+1)} - z^{(k)}| \quad \Delta z \leftarrow |z^{(k+1)} - z^{(k)}|$$

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

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

5. Update λ adaptively:

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

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

$$\lambda^{(k+1)} \leftarrow \lambda^{(k)} \cdot (1 + \beta_\lambda \cdot \frac{\Delta HitRate}{\Delta PollutionRate})$$

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

6. Check for convergence:

if

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

Break (converged)

If both β and z have converged, stop the algorithm

End for

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

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 شبیه سازی ها و تحلیل ها، از ترکیب دو ابزار مکمل استفاده شده است تا هم جنبه های تحلیلی و عددی و هم ارزیابی های شبکه ای به صورت دقیق پوشش داده شود. پیاده سازی الگوریتم پیشنهادی و تحلیل رفتار بهینه سازی پارامتر β در محیط MATLAB 2020Rb انجام شده است؛ این محیط با دقت 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 شبکه های NDN، از شبیه ساز تخصصی ndnSIM، مبتنی بر NS-3، بهره گرفته شد. این ابزار امکان شبیه سازی دقیق تعامل گره ها، حافظه نهان و شرایط حمله را فراهم می کند. سخت افزار مورد استفاده شامل پردازنده Intel Core i7 با فرکانس ۲.۶۰ گیگاهرتز و ۸ گیگابایت رم تحت سیستم عامل Windows 10 بوده است. این ترکیب ابزاری، توازن مؤثری میان تحلیل عددی دقیق و ارزیابی عملیاتی الگوریتم در محیط شبکه ای ایجاد کرده است.

۴-۱- معیارهای ارزیابی

در این پژوهش، برای ارزیابی عملکرد روش پیشنهادی، دو معیار اصلی به کار گرفته شده اند که نقش کلیدی در تحلیل و مقایسه الگوریتم ها دارند. اولین معیار، نرخ اصابت حافظه نهان^۱ (CHR)، نشان دهنده درصد درخواست های پاسخ داده شده مستقیماً از کش است و معیار اساسی برای سنجش کارایی سیستم کش محسوب می شود؛ چرا که افزایش این نرخ منجر به کاهش تأخیر در تحویل محتوا و بهبود تجربه کاربر می شود.

$$CHR\% = \frac{\text{Number of cache hits}}{\text{Total number of content requests}} \times 100 \quad (11)$$

در این رابطه Number of cache hits تعداد درخواست هایی که از حافظه نهان پاسخ داده شده اند (تعداد اصابت ها به کش) و Total number of content requests تعداد کل درخواست های محتوای کاربران (صرف نظر از اینکه از کش پاسخ داده شده اند یا نه) است.

دومین معیار، درصد آلودگی حافظه نهان^۲ (CPR)، شاخصی مهم در شناسایی و کاهش محتوای غیر مفید و حملات آلودگی کش است و بیانگر توانایی الگوریتم در حفظ کیفیت داده های ذخیره شده می باشد.

$$CPR\% = \frac{\text{Number of polluted contents in the cache}}{\text{Total number of cached contents}} \times 100 \quad (12)$$

در این رابطه Number of polluted contents in the cache تعداد داده های آلوده یا غیر محبوب موجود در حافظه نهان و Total number of cached contents تعداد کل داده هایی که در حافظه نهان ذخیره شده اند.

۴-۲- توپولوژی ها و سناریوها

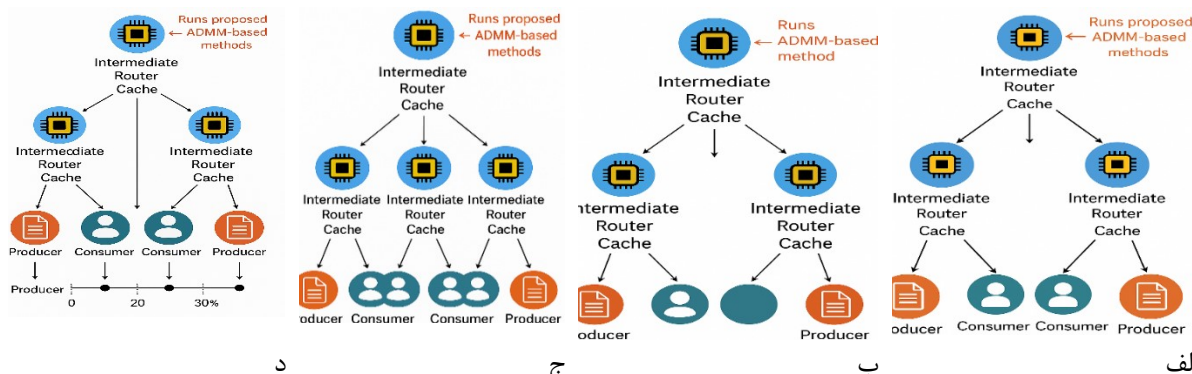
در این پژوهش، چهار نوع توپولوژی شبکه شامل سلسله مراتبی، ساده، پیشرفته و مش (شکل ۳) طراحی شده اند تا عملکرد الگوریتم پیشنهادی به ویژه الگوریتم ADMM و نسخه بهبود یافته آن ارزیابی شود. در توپولوژی سلسله مراتبی (شکل الف)، شبکه شامل سه لایه است: لایه اول که گره های تولیدکننده محتوا را در بر می گیرد، لایه دوم که گره های میانی با حافظه های نهان هستند، و لایه سوم که گره های مصرف کننده در آن قرار دارند. در این توپولوژی، سه سناریو طراحی شده است: سناریوی اول شبکه پایدار است که به بررسی تأثیر ظرفیت کش در سه مقدار مختلف می پردازد. سناریوی دوم به بررسی تاب آوری الگوریتم در برابر حملات آلودگی کش (CPA) با افزایش تعداد درخواست های مخرب پرداخته است. در نهایت، سناریوی سوم تغییرات بار ترافیکی را مورد بررسی قرار می دهد که در آن نرخ درخواست ها به طور دینامیک از ۵ به ۲۰ درخواست در ثانیه تغییر می کند. در توپولوژی ساده (شکل ب)، شبکه شامل سه لایه است: گره های تولید کننده، مصرف کننده و میانه با حافظه نهان. در این توپولوژی نیز سه سناریو طراحی شده است: سناریوی شبکه پایدار که عملکرد الگوریتم در شرایط عادی بررسی می شود، حمله

¹ Cache Hit Rate

² Cache Pollution Rate

آلودگی کش 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 (شکل ج)، ساختار پیچیده‌تری از سلسله‌مراتب با به‌روزرسانی پارامترهای الگوریتم ADMM در گره‌های میانی به صورت توزیع شده 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 کش (CPA)، مقاومت الگوریتم در برابر حملات آلودگی کش بررسی می‌شود و در تغییرات بار ترافیکی (پویا)، توانایی الگوریتم در پاسخ به تغییرات دینامیکی بار ترافیکی ارزیابی می‌شود. در توپولوژی مش (شکل د)، گره‌ها به صورت 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 غیرمتعادل می‌پردازد، قطع برخی لینک‌ها که ارزیابی پایداری و تاب‌آوری الگوریتم در برابر خرابی‌های شبکه را شامل می‌شود، و حمله هدفمند توزیع شده که به بررسی مقاومت الگوریتم در برابر حملات توزیع شده آلودگی کش می‌پردازد.



شکل (۳) توپولوژی‌های بررسی شده (الف) سلسله‌مراتبی، (ب) ساده، (ج) پیشرفته، (د) مش

Topologies examined a) hierarchical, b) simple, c) advanced, d) mesh

این توپولوژی‌ها و سناریوها به‌طور 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 شبیه‌سازی شده در این پژوهش آورده شده است.

جدول (۲) بررسی جامع توپولوژی‌ها و سناریوهای مختلف شبیه‌سازی شده

Table (2) comprehensive review of different simulated topologies and scenarios

توپولوژی شبکه	سناریو	هدف ارزیابی	شرح
توپولوژی سلسله‌مراتبی	شبکه پایدار	ارزیابی حساسیت الگوریتم به ظرفیت حافظه کش و بررسی نرخ اصابت کش و آلودگی کش	بررسی عملکرد الگوریتم در شرایط عادی و بدون حمله، با تنظیم ظرفیت کش در سه مقدار مختلف (۵۰۰، ۱۰۰۰، ۲۰۰۰ محتوا)
	حمله آلودگی کش (CPA)	ارزیابی مقاومت الگوریتم در برابر حملات آلودگی کش و حفظ کیفیت سرویس	بررسی تاب‌آوری الگوریتم در برابر حملات آلودگی کش 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 کش	بررسی توانایی الگوریتم در مدیریت تغییرات ناگهانی یا تدریجی در تعداد کاربران و نرخ درخواست‌ها (از ۵ به ۲۰ درخواست در ثانیه)
توپولوژی ساده	شبکه پایدار	ارزیابی نرخ اصابت کش (CHR) و آلودگی کش (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 توزیع Zipf

	حمله آلودگی کش (CPA)	بررسی مقاومت الگوریتم در برابر حملات آلودگی کش و حفظ نرخ اصابت کش و کاهش آلودگی کش	بررسی عملکرد الگوریتم در برابر درخواست های جعلی و غیرمحبوب ارسال شده توسط مهاجمان به حافظه های نهان
	تغییرات بار ترافیکی (پویا)	بررسی توانایی الگوریتم در تطبیق با تغییرات بار ترافیکی و تنظیم خودکار پارامترهای β ، λ و ρ	بررسی تغییرات ناگهانی یا تدریجی در تعداد کاربران و نرخ درخواست ها 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 کش و کاهش آلودگی کش	بررسی عملکرد الگوریتم در محیط بدون حمله با ساختار پیچیده تر و چندلایه (تولیدکننده، حافظه های نهان، مصرف کننده)
	حمله آلودگی کش (CPA)	ارزیابی مقاومت الگوریتم در برابر حملات آلودگی کش و حفظ کیفیت کش و بهینه سازی پارامترها	بررسی تاب آوری الگوریتم در برابر حملات آلودگی کش با درخواست های جعلی و داده های غیرمحبوب ارسال شده توسط مهاجمان
	تغییرات بار ترافیکی (پویا)	سنجش انطباق پذیری 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 درخواست ها و ارزیابی قدرت الگوریتم در تطبیق خود با شرایط متغیر شبکه
توپولوژی مش	ازدحام موضعی و مسیرهای غیرتعادلی	ارزیابی قدرت الگوریتم در توزیع متوازن بار و حفظ تعادل کش ها و کاهش تأخیر در تحویل محتوا	بررسی بار ترافیکی سنگین در بخش های خاص شبکه و ارزیابی توانایی الگوریتم در مدیریت ازدحام و توزیع بار به طور هوشمندانه میان مسیرهای جایگزین
	قطع برخی لینک ها	ارزیابی انعطاف الگوریتم در شرایط خرابی لینک ها و حفظ کیفیت سرویس و تحویل محتوا بدون اختلال	شبیه سازی خرابی لینک ها و بررسی عملکرد الگوریتم در شرایط قطع یا اختلال لینک ها و شناسایی مسیرهای جایگزین برای حفظ عملکرد کش
	حمله توزیع شده آلودگی کش	سنجش مقاومت الگوریتم در برابر حملات توزیع شده آلودگی کش و توانایی آن در حفظ کیفیت کش و بهینه سازی محتوا	بررسی عملکرد الگوریتم در برابر حملات هماهنگ و توزیع شده که به صورت هم زمان از مسیرهای مختلف درخواست های جعلی و غیرمحبوب به شبکه ارسال می شود

۴-۳- ارزیابی در توپولوژی سلسله مراتبی

در این پژوهش، عملکرد الگوریتم های ADMM و نسخه بهبود یافته آن (ADMM-Enhanced) در سه سناریو مختلف شبکه در توپولوژی سلسله مراتبی بررسی شد. در سناریو شبکه پایدار (جدول ۳)، الگوریتم های ADMM و ADMM-Enhanced نسبت به روش های کلاسیک مانند LFU-DA و LRU عملکرد بهتری داشتند، با نرخ اصابت کش بالاتر و درصد آلودگی کش پایین تر. در سناریوی حمله آلودگی کش (جدول ۴)، این الگوریتم ها توانستند مقاومت بیشتری در برابر حملات نشان دهند و آلودگی کش را کاهش داده و نرخ اصابت کش را حفظ کنند، در حالی که روش های کلاسیک آسیب پذیرتر بودند. در سناریوی تغییرات بار ترافیکی (جدول ۵)، الگوریتم های ADMM و ADMM-Enhanced با توانایی تطبیق پذیری بالا، در شرایط پویا عملکرد مطلوبی داشتند و نرخ اصابت کش را حفظ کرده و آلودگی کش را کنترل کردند. به طور کلی، این الگوریتم ها در تمام سناریوها نسبت به روش های مرسوم عملکرد برتری نشان داده اند و به عنوان یک راهکار مؤثر برای مدیریت حافظه های نهان در شبکه های داده محور پیشنهاد می شوند.

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

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

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

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

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

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

الگوریتم	Top 50 نرخ اصابت	Top 100 نرخ اصابت	درصد آلودگی کش (CPR)	نرخ اصابت کش (CHR)
PFP-DA	70.00%	60.00%	14.00%	18.00%
LFU-DA	52.00%	45.00%	28.00%	11.00%
LRU	48.00%	40.00%	30.00%	9.00%
ADMM	75.00%	63.00%	11.00%	20.00%
ADMM-Enhanced	77.00%	65.00%	9.00%	22.00%

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 کش بالاتری داشته باشند و همزمان درصد آلودگی کش را در سطح پایین‌تری حفظ کنند. در حالی که روش‌های دیگر مانند LFU-DA و LRU نرخ اصابت کش کمتری داشتند و درصد آلودگی کش در آن‌ها بالاتر بود. در سناریوی حمله آلودگی کش (CPA) (جدول ۷)، الگوریتم‌های ADMM و ADMM-Enhanced مقاومت بهتری نشان دادند و توانستند با وجود حمله، آلودگی کش را به میزان چشمگیری کاهش دهند و نرخ اصابت کش را در سطح بالایی حفظ کنند. در مقابل، الگوریتم‌های کلاسیک مانند LFU-DA و LRU در برابر حملات آسیب‌پذیر بودند و با افزایش آلودگی کش 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 کش را حفظ کرده و آلودگی کش را در سطح مطلوبی کنترل کنند. به طور کلی، نتایج نشان می‌دهند که الگوریتم‌های ADMM و ADMM-Enhanced به دلیل 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)

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

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

الگوریتم	Top 50 نرخ اصابت	Top 100 نرخ اصابت	درصد آلودگی کش (CPR)	نرخ اصابت کش (CHR)
PFP-DA	73.00%	61.00%	16.00%	18.50%
LFU-DA	49.00%	40.00%	35.00%	11.50%
LRU	33.00%	27.00%	37.00%	7.50%

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

۴-۵- نتایج ارزیابی برای توپولوژی پیشرفته

الگوریتم های ADMM و نسخه پیشرفته آن (ADMM-Enhanced) در سه سناریو مختلف توپولوژی پیشرفته ارزیابی شدند. در سناریوی شبکه پایدار (جدول ۹)، الگوریتم های پیشنهادی عملکرد بهتری نسبت به روش های کلاسیک مانند LRU و LFU-DA داشتند. این الگوریتم ها توانستند نرخ اصابت کش بالاتری را ثبت کنند و در عین حال، درصد آلودگی کش را به طور قابل توجهی کاهش دهند. همچنین، در این سناریو، نرخ اصابت برای محتوای پرتعدادار Top 100 و Top 50 در الگوریتم های ADMM و نسخه پیشرفته آن بالاترین میزان را داشت. در سناریوی حمله آلودگی کش (CPA) (جدول ۱۰)، الگوریتم های ADMM و ADMM-Enhanced به طور مؤثری آلودگی کش را کاهش دادند و توانستند تعادل مناسبی بین نرخ اصابت و مقاومت در برابر آلودگی ایجاد کنند. در حالی که روش های کلاسیک نظیر LRU و LFU-DA با کاهش قابل توجه در نرخ اصابت و افزایش آلودگی کش مواجه شدند. در سناریوی تغییرات بار ترافیکی (پویا) (جدول ۱۱)، الگوریتم های ADMM و ADMM-Enhanced عملکرد پایدار و تطبیق پذیری خوبی در شرایط متغیر و پویا از خود نشان دادند. این الگوریتم ها توانستند نرخ اصابت کش را در سطح مطلوب حفظ کرده و آلودگی کش را کنترل کنند. در مقابل، روش های مرجع در شرایط تغییرات بار ترافیکی ضعیف عمل کردند. به طور کلی، نتایج این سناریوها نشان می دهند که الگوریتم های ADMM و ADMM-Enhanced در تمامی شرایط از نظر نرخ اصابت کش، مقاومت در برابر آلودگی و مدیریت محتواهای پربازدید عملکرد بهتری نسبت به روش های مرجع دارند و به عنوان گزینه ای ایده آل برای شبکه های NDN پیشنهاد می شوند.

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

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

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

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

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

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

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

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

الگوریتم	Top 50 نرخ اصابت	Top 100 نرخ اصابت	(CPR) درصد آلودگی کش	(CHR) نرخ اصابت کش
PFP-DA	74.00%	63.00%	10.00%	21.00%
LFU-DA	56.00%	47.00%	24.00%	14.00%

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 کش 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 کش مواجه شدند، در حالی که الگوریتم‌های مبتنی بر ADMM با ساختار تطبیقی خود، توانستند کارایی بالاتری را از نظر ذخیره‌سازی محتوای پرتعداد و کاهش داده‌های کم‌ارزش نشان دهند. در سناریوی حمله آلودگی کش (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 الگوریتم‌های ADMM، 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 کش و تصمیم‌گیری توزیع‌شده، توانستند در شرایط متغیر بار، عملکرد مطلوبی از خود نشان دهند و کیفیت کش را حفظ کنند، در حالی که روش‌های غیرهوشمند دچار افت کارایی شدند. این نتایج نشان می‌دهند که الگوریتم‌های پیشرفته مانند ADMM، به‌ویژه نسخه پیشرفته آن، در تمامی شرایط از جمله شبکه‌های پایدار، حملات آلودگی کش 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	Top 50 نرخ اصابت	Top 100 نرخ اصابت	CPR درصد آلودگی کش	CHR نرخ اصابت کش
PFP-DA	77.00%	67.00%	11.00%	20.00%
LFU-DA	63.00%	53.00%	21.00%	13.00%
LRU	57.00%	47.00%	27.00%	10.00%
ADMM	79.00%	69.00%	9.00%	22.00%
ADMM-Enhanced	81.00%	71.00%	8.00%	24.00%

جدول (۱۳) سناریو ۲ توپولوژی مش: حمله آلودگی کش (CPA)

Table (13) Mesh Topology Scenario 2: CPA

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

جدول (۱۴) سناریو ۳ توپولوژی مش: تغییر بار ترافیکی (پویا)

Table (14) scenario 3 mesh topology: traffic load change (dynamic)

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

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 ۰.۰۵) باعث بهبود نرخ اصابت کش و کاهش آلودگی کش 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 ρ و 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. با β بهینه (۰.۰۵) توانسته آلودگی کش 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 50 و Top 100 داشت. در سناریو دوم، جدول (۱۶)، که حمله با بار مهاجم ۵٪ بررسی شد، ADMM و ADMM-Enhanced به دلیل استفاده از تنظیم تطبیقی β توانستند نرخ اصابت کش و نرخ اصابت برای محتوای پرتعداد 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 کش داشته باشند. به‌ویژه 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)

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

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

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

۴-۸- مقایسه کیفی توپولوژی‌ها

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 (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
	ADMM-Enhanced	عالی	عالی	عالی	عالی	بالا
پیشرفته	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
	ADMM-Enhanced	عالی	عالی	عالی	عالی	بالا
مش fault – tolerance	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
	ADMM-Enhanced	عالی	عالی	عالی	عالی	بالا
مقاله پایه – بدون حمله ($\beta=0.05$)	PFP- β -DA	خوب	عالی	عالی	خوب	متوسط
	ADMM	عالی	عالی	عالی	عالی	بالا
	ADMM-Enhanced	عالی +	عالی +	عالی +	عالی +	بسیار بالا
مقاله پایه – حمله ۳۰٪ مهاجم	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، با هدف مقابله هوشمندانه و تطبیق پذیر با چالش های مدیریت کش در معماری شبکه های داده های نام گذاری شده (NDN)، الگوریتمی مبتنی بر ADMM و نسخه بهبود یافته آن (ADMM-Enhanced) طراحی و ارزیابی شد. این الگوریتم با تنظیم پویا و توزیع شده پارامتر کلیدی β ، و همچنین بهینه سازی همزمان λ و ρ ، توانست در سناریوهای متنوع شامل شبکه پایدار، حملات آلودگی کش و تغییرات شدید بار، عملکردی برتر نسبت به روش های مرجع مانند LRU-DA، LRU-DA و حتی PFP-DA از خود نشان دهد. نتایج شبیه سازی در توپولوژی های مختلف از ساده تا مش-نشان داد که ADMM-Enhanced در حفظ نرخ بالای اصابت کش و کاهش چشمگیر آلودگی، به ویژه در شرایط ناپایدار و مهاجم پذیر، بسیار موفق بوده و از نظر عددی و عملی به عنوان یک راهکار مقیاس پذیر و مقاوم برای مدیریت هوشمند کش در نسل آینده شبکه های داده محور قابل اتکا 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. Bouzeffrane, "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. Cograane, 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.

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

Abstract

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

Correspondence

Mohammad Reza Khayam bashi, professor,
Faculty of Computer Engineering - University of Isfahan - Isfahan – Iran
Email: m.r.khayyambashi@eng.ui.ac.ir

Received: 23 October 2023

Revised: 18 January 2024

Accepted: 12 February 2024

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