



پنجمین کنفرانس ملی مهندسی برق و سیستم های هوشمند ایران - دانشگاه آزاد اسلامی واحد نجفآباد - ۸ و ۹ اسفند ۱۳۹۷

ارائه یک مدل پیش بینی ترافیک شبکه های نرم افزار محور با استفاده از فیلتر کالمن

علی مسیبه^۱، فرهاد فغانی^{۲*}

^۱دانشکده مهندسی برق، واحد نجف آباد، دانشگاه آزاد اسلامی، نجف آباد، ایران.

^{۲*}Ali_mosayebi60@yahoo.com^۱, Faghani@iaun.ac.ir^۲

چکیده - امروزه پیش بینی ترافیک شبکه ها بسیار مورد توجه و علاقه پژوهشگران و صنعت است. زیرا پیش بینی شرایط بعدی شبکه موجب مدیریت و تصمیم گیری بهینه، افزایش کارآمدی و در نهایت رضایت کاربر خواهد شد. از سویی با گسترش شبکه های نرم افزار محور زمینه های کاربردی بسیار زیادی جهت بکارگیری و بهبود سیستم های مهندسی ترافیک بوجود آمده است. از آنجا که جریان یکی از واحدهای ارائه ترافیک در شبکه نرم افزار محور است؛ در این مقاله، پیش نویس یک مدل مبتنی بر فیلتر کالمن برای پیش بینی ترافیک شبکه های نرم افزار محور ارائه شده است. برای پیش بینی نرخ جریان ترافیک برای بازه های زمانی بعدی، از حجم ترافیک در واحد زمان و یا نرخ جریان ترافیکی مشاهده شده در یک فاصله زمانی خاص و با استفاده از محاسبات بازگشتی پارامترهای مربوطه از داده های ترافیک ورودی استفاده شده است، که می تواند در شبکه *OpenFlow* استفاده شود. پارامترهای مربوطه عبارتند از: کوواریانس نوین فرآیند، کوواریانس نوین اندازه گیری، بهره فیلتر کالمن، حالت پیشین و پسین و کواریانس تخمین. در نهایت این مدل با استفاده از مجموعه داده با کمترین، متوسط و بیشترین اتوکورلیشن آزمایش شد.

کلید واژه: پیش بینی، شبکه های نرم افزار محور، فیلتر کالمن، مهندسی ترافیک

۱- مقدمه

پیش بینی ترافیک شبکه های نرم افزار محور ارائه شده است. حجم ترافیک در واحد زمان و یا نرخ جریان ترافیکی مشاهده شده در یک فاصله زمانی خاص، برای پیش بینی نرخ جریان ترافیک برای بازه های زمانی بعدی با استفاده از محاسبات بازگشتی پارامترهای مربوطه از داده های ترافیک ورودی استفاده شده است.

۲- شبکه های نرم افزار محور

در گذشته، ایجاد نوآوری و مدیریت شبکه بسیار سخت بود زیرا هر دستگاه دارای منطق کنترل خاص خود و کاملاً وابسته به فروشنده بود. برای شکستن این محدودیت، *SDN* بر اساس *OpenFlow*، به عنوان یک الگوی تکاملی در شبکه های مرکز داده، پیشنهاد شد [۲]. این ساختار شامل سه بخش عمده است: سطح کنترل، سطح داده و برنامه کاربردی (شکل ۱). لایه کنترل از طریق رابط شمالی (*northbound interface*) با لایه برنامه و بوسیله رابط جنوبی (*southbound interface*) با سوئیچ *SDN* در سطح داده در ارتباط است.

در شبکه های انتقال داده، تجزیه و تحلیل و پیش بینی ترافیک یک رویکرد پیشگیرانه برای تضمین امنیت شبکه، قابل اطمینان بودن، اجتناب از ازدحام، افزایش سرعت شبکه و کاملاً مرتبط با کیفیت شبکه است [۱]. امروزه با گسترش شبکه های نرم افزار محور یا *SDN* زمینه های کاربردی بسیار زیادی جهت بکارگیری و بهبود سیستم های مهندسی ترافیک بوجود آمده است. برتریهای *SDN* بر شبکه های سنتی، بخاطر وجود یک نهاد مرکزی در شبکه است که در هر لحظه می تواند یک دید کلی از شرایط فعلی و تغییرات نرم افزاری و رفتار مسیریابی از تمام تجهیزات شبکه در اختیار داشته باشند. با این حال روش های مهندسی ترافیک فعلی توسعه یافته برای *SDN*، به دلیل آنکه آنها فقط در برابر شرایط فعلی شبکه واکنش نشان می دهند، نمی توانند کاملاً از ازدحام جلوگیری کنند. قابلیت پیش بینی ترافیک شبکه، تا حد زیادی به سیستم مهندسی ترافیک اجازه تغییر مسیر ترافیک شبکه قبل از بوجود آمدن شرایط ازدحام را می دهد. در این پروژه، یک مدل مبتنی بر فیلتر *Kalman* برای

۳- فیلتر کالمن

فیلتر کالمن اساساً مجموعه‌ای از معادلات ریاضی است که یک پیش‌بینی کننده-اصلاح کننده را اجرا می‌کند، این به این معنی است که اگر تعدادی از شرایط احتمالی برآورده شود، کوواریانس خطای تخمینی به حداقل می‌رسد. در واقع با استفاده از یک سری مشاهدات وابسته به حالت سیستم، حالت سیستم تخمین زده می‌شود. در واقع برای یک فیلتر کالمن گسسته، حالت فعلی سیستم به صورت زیر بدست می‌آید:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (1)$$

که در آن x_k حالت سیستم در لحظه k ام و u_k ورودی سیستم در لحظه k ام و w_k نویز یا خطای حالت سیستم می‌باشد. همچنین بردار مشاهدات را می‌توان به صورت زیر تعریف نمود:

$$z_k = Hx_k + v_k \quad (2)$$

که در آن، v_k نویز یا خطای اندازه‌گیری است. در این معادلات فرض بر آن است که w_k و v_k مستقل از یکدیگر، سفید و با تابع چگالی احتمال نرمال می‌باشند:

$$p(w) \sim N \quad (3)$$

$$(0, Q)$$

$$p(v) \sim N \quad (4)$$

$$(0, R)$$

برای معادلات مرتبط با فیلتر کالمن، ما با دو دسته از روابط روبرو هستیم: معادلات مربوط به پیش‌بینی و معادلات مربوط به تصحیح. این دو دسته از معادلات به یکدیگر وابسته بوده و از روی یکدیگر به روز می‌شوند. برای حل معادلات مربوط به پیش‌بینی (بروزرسانی زمان) کافی است که مقادیر اولیه تخمین \hat{x}_{k-1} و P_{k-1} را مشخص نماییم و در معادلات زیر جایگزین می‌کنیم:

$$\hat{x}_k^- = A\hat{x}_{k-1} \quad (5)$$

$$P_k = AP_{k-1}A^T + Q \quad (6)$$

نتایج بدست آمده در این قسمت را در معادلات مربوط به تصحیح، جایگزین می‌کنیم. با جایگذاری روابط بدست آمده از قسمت قبل در روابط زیر، معادلات مربوط به تصحیح بروز میشود (بروزرسانی اندازه گیری).

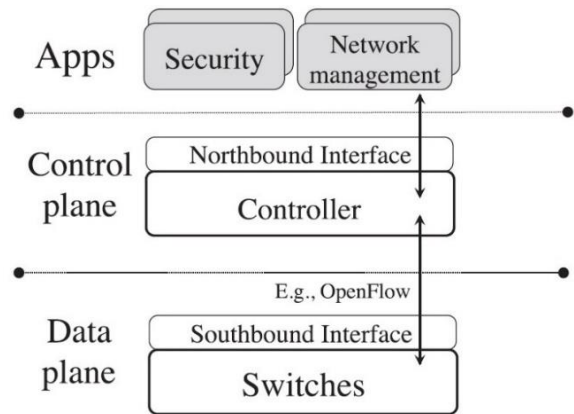
$$K_k = P_k H_k^T (H_k P_k H_k^T + R)^{-1} \quad (7)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k^T \hat{x}_k^-) \quad (8)$$

$$P_k = (I - K_k H_k) P_k \quad (9)$$

سپس نتایج بدست آمده از این قسمت را در معادلات مربوط به پیش‌بینی اعمال کرده و این روند را ادامه می‌دهیم [۶].

سطح کنترل، تجهیزات انتقال را از طریق کنترل کننده و با توجه به دستورالعمل‌ها یا ورودی‌های جریان مدیریت می‌کند، تا بسته‌ها را ارسال و به هدف خاص برنامه نهایی دست‌یابند. سطح داده بر اساس پروتکل مشترک (به عنوان مثال *OpenFlow*) با کنترل کننده در ارتباط است. دستگاه‌هایی که در آن *OpenFlow* فعال است را می‌توان توسط کنترل کننده‌ها از طریق کانال امن کنترل کرد [۳].



شکل ۱: ساختار شبکه‌های نرم افزار محور [۳].

در شبکه سخت‌افزاری سنتی، سوئیچ‌ها معمولاً سیستم‌های بسته هستند که دارای لایه‌های کنترل و داده خود هستند و از رابط‌های کنترلی خاص خود پشتیبانی می‌کنند. بنابراین، استقرار خدمات و پروتکل‌های جدید (و حتی پروتکل‌های موجود در نسخه‌های جدید) یک چالش است، زیرا همه سوئیچ‌ها باید به روز یا جایگزین شوند. در مقابل، در *SDN*، سوئیچ‌ها به‌خاطر حذف قسمت کنترل، به دستگاه‌های حمل و نقل داده ساده بدل می‌شوند، که دیگر نیازی به شناسایی و پردازش هزاران استاندارد و پروتکل را نخواهند داشت و یک کنترل کننده متمرکز مکانیسم کنترل شبکه را ایجاد می‌کند [۴]. برتری‌های *SDN* بر شبکه‌های سنتی در زمینه مهندسی ترافیک بخاطر وجود همین نهاد کنترل کننده مرکزی در شبکه است، که با یک دسترسی و دید کلی و کامل به حالت فعلی شبکه و سوئیچ‌ها، می‌تواند آمار جریان سطح لینک‌های شبکه را جمع‌آوری کند و همزمان و در هر لحظه - بصورت نرم‌افزاری سوئیچ‌ها را از طریق کنترل کننده برنامه‌ریزی و رفتار مسیریابی تجهیزات شبکه را تغییر دهد و به این شکل ارائه پروتکل‌ها و سرویس‌های جدید بسیار ساده‌تر شده است [۵].

از (ب) می‌توان نتیجه‌گیری شود که نرخ جریان ترافیک در هر مرحله زمانی خاص یک ترکیب خطی از جریان‌های ترافیکی در مراحل زمان قبلی است. به عبارت دیگر، مجموع نرخ‌های قبلی جریان ترافیک با ضریب مربوطه، جریان ترافیک را برای مرحله زمان فعلی ارائه می‌دهند.

بنابراین، اگر trf_k نرخ جریان ترافیک را در مرحله k ام نشان دهد و C_k نشان دهنده ضریب مربوطه باشد، پس از (ب) میتوان نتیجه‌گیری شود:

$$(10)$$

$trf_k = trf_{k-1}c_{k-1} + \dots + trf_{k-n}c_{k-n} + v_k$
که v_k سهم نوین در مرحله k و n نشان‌دهنده تعداد جریان های ترافیکی قبلی است که در نظر گرفته می‌شود. سمت راست معادله بالا را با تعاریف زیر به فرم ماتریسی بیان می‌کنیم:

$$(11)$$

$$TRF_k = [trf_{k-1}, trf_{k-2}, trf_{k-3}, \dots, trf_{k-n}]$$

$$(12)$$

$$C_k = [c_{k-1}, c_{k-2}, c_{k-3}, \dots, c_{k-n}]^T$$

بنابراین معادله (۱۰) به شکل زیر بازنویسی می‌شود:

$$(13)$$

$trf_k = TRF_k C_k + v_k$
با مقایسه معادله‌های (۱۳) با (۲)، مشاهده می‌شود که راه مناسب برای مدل‌سازی سیستم، فرض موارد زیر است:

$$trf_k = Z_k$$

خروجی اندازه‌گیری شده:

$$TRF_k = h$$

ماتریس بهره خروجی:

$$C_k = x_k$$

حالت سیستم:

بدین ترتیب با سیستم مدل‌سازی شده، معادله خروجی فیلتر یا همان معادله (۲) ممکن است به صورت زیر باشد

$$tff_k = TRF_k C_k$$

(۱۴)

که در آن tff_k نرخ جریان ترافیکی خروجی تخمین زده شده، و C_k تخمین حالت پیشین را نشان می‌دهند.

پس از بدست آوردن جریان ترافیک تخمینی خروجی یا

tff_k ، ترافیک خروجی اندازه‌گیری شده یعنی trf_k بدست می‌آید. و از trf_k برای به روز رسانی ماتریس خروجی

TRF_k استفاده می‌شود؛ بنابر این خواهیم داشت:

$$(15)$$

$$TRF_{k+1} = [trf_k, trf_{k-1}, \dots, trf_{k-n+1}]$$

تمرکز این مقاله بر ایجاد یک مدل فیلتر کالمن مناسب برای پیش‌بینی سرعت جریان ترافیک در یک SDN است. در این بخش جزئیات روند توسعه یک مدل کارآمد را برای پیش‌بینی سرعت جریان ترافیک بیان می‌کنیم همانطور که گفتیم برای پیش‌بینی نرخ جریان ترافیک برای بازه‌های زمانی بعدی، از حجم ترافیک در واحد زمان و یا نرخ جریان ترافیکی مشاهده شده در یک فاصله زمانی خاص استفاده می‌کنیم.

باتوجه به مفهوم طراحی *OpenFlow*، در هر دستگاه *OpenFlow* چندین شمارنده وجود دارد که اطلاعات بسته‌های پردازشی از جمله قوانین انتقال (*forwarding rules*) و مشخصات جریان (*flow entries*) همه سوئیچ‌ها را در کنترل-کننده نگهداری می‌کند. معمولاً اطلاعاتی از قبیل شمارش بسته، شمارش بیت و تعداد بسته کاهش یافته در هر دوره زمانی در جدول جریان (*Flow Table*) ذخیره می‌شوند. در سیستم طراحی شده در این مقاله یک تابع با پرس و جو از شمارنده، نرخ جریان ترافیکی در بازه دلخواه ما را دریافت می‌کند. کفایت یک درخواست به کنترل‌کننده ارسال شود. تا کنترل‌کننده به طور صحیح آمار جریان را از سوئیچ‌های مختلف ارائه دهد.

برای مثال، در یک فاصله زمانی که توسط مدیر ارائه می‌شود، اطلاعات پورت مثل: نرخ دریافت و ارسال بیت؛ نرخ ارسال و دریافت بسته؛ اندازه متوسط بسته و شماره بسته از دست‌رفته؛ و آمار جریان مثل: نرخ بسته و نرخ بایت در هر جریان، قابل دریافت است [۷].

۴-۱ معرفی مدل سیستم برای فیلتر کالمن

تجزیه و تحلیل در این بخش ابتدا بر روی یک لینک شبکه، و سپس برای پوشش چندین لینک شبکه گسترش می‌یابد. مدل کالمن برای یک لینک تنها به عنوان یک مدل اسکالر، سپس با گسترش به چندین لینک با استفاده از ماتریس‌ها مورد بررسی قرار می‌گیرد. به دلایل زیر در این مقاله با پیش‌بینی سرعت جریان ترافیکی کوتاه مدت شروع می‌کنیم:

الف: برای پیش‌بینی کوتاه مدت، تغییر متغیر حالت را می‌توان به عنوان یک فرایند صاف در نظر بگیریم.

ب: یک رابطه خطی بین نرخ جریان ترافیک برای گام زمان فعلی و نرخ جریان ترافیک برای گام زمان قبلی فرض شده است.

شود. با مقایسه این دو مقدار؛ اگر $rms_current$ کمتر یا برابر $rms_acceptable$ باشد، پس فیلتر یک نتیجه رضایت بخش تولید کرده است، و پیش‌بینی فعلی به عنوان یک پیش‌بینی موفقیت آمیز شمرده می‌شود. در طی پیشرفت تکرارهای فیلتر، بازده فیلتر به صورت نسبت تعداد پیش‌بینی‌های موفق به تعداد کل پیش‌بینی‌ها محاسبه می‌شود. با اجرای فیلتر، میانگین خطای مربع بین پیش‌بینی و مقادیر واقعی نظارت می‌شود. بازده نهایی، بازده اندازه‌گیری شده برای یک خطای میانگین مربع (mse) ثابت شده خواهد بود.

۵- پیاده‌سازی مدل

پیاده‌سازی مدل توسعه‌یافته در بخش ۴ توسط "نرم‌افزار جاوا" و با پیروی از الگوریتم فیلتر کاملن که در فرم ماتریسی معادلات (۱) تا (۷) ارائه شده است انجام شد. هدف اولیه اجرای مدل در جاوا، بررسی صحت رویکرد توصیف شده برای اجرای این فیلتر می‌باشد. در توسعه آینده این مدل، ما آنرا در محیط مینینت به کار خواهیم گرفت. جایی که پارامترها می‌توانند در زمان واقعی استفاده و اثر مستقیم آنها بررسی و رصد شوند.

شبیه‌ساز *Mininet* برنامه‌ای است که می‌تواند شبکه‌ای مجازی را ایجاد کرده و کرنل‌های واقعی و کدهای برنامه و سویچ‌های شبکه را بر روی یک ماشین (ماشین مجازی، ابر یا سیستم واقعی) به اجرا درآورد. این نرم‌افزار به دلیل امکان تعامل پذیری توسط خط فرمان و یا *API* می‌تواند در راستای توسعه، آموزش و تحقیق استفاده شود. همچنین این نرم‌افزار قابلیت‌های گسترده‌ای به منظور تست، ارزیابی و توسعه ابزارهای مبتنی بر *SDN* و *OpenFlow* خواهد داشت.

۵-۱- مقدار دهی اولیه:

این سیستم با در نظر گرفتن تعداد m لینک و تعداد n نرخ جریان ترافیکی قبلی، به شکل زیر مقداردهی اولیه شده، و مراحل با مقدار اولیه $k=1$ شروع و تا رسیدن به نتیجه نهایی ادامه یافت.

*ماتریس ورودی متغیر U ، برابر صفر

* \hat{X}_k یا \hat{C}_k یک ماتریس $n \times I$ با مقادیر کوچک I/n

*ماتریس بهره حالت یا A یک ماتریس واحد با ابعاد $n \times n$

*ماتریس بهره ورودی یا B ، برابر صفر

* H یا TRF_k یک ماتریس $m \times n$ با مقادیر کوچک مثلاً

۰,۰۰۹

برای دستیابی به هدف فیلتر پیش‌بینی میزان جریان ترافیک خروجی برای گام بعدی، معادله (۱۴) می‌تواند مورد استفاده قرار گیرد:

$$tff_{k+1} = \quad (16)$$

$$TRF_{k+1} \hat{C}_k$$

که در آن \hat{C}_k تخمین حالت پسین می‌باشد.

۲-۲-۴- لینک‌های چندگانه

به منظور گسترش تجزیه و تحلیل بخش قبلی برای پوشش چندین لینک، با شبکه به عنوان یک ماتریس رفتار می‌شود. پارامترهای مختلف بر حسب تعداد لینک‌ها در شبکه و تعداد جریان‌های ترافیکی قبلی مورد توجه قرار می‌گیرند. برای یک شبکه با m لینک، یک ماتریس خروجی $m \times I$ برای نگه داشتن مقادیر خروجی هر یک از m لینک تعریف می‌شود. به همین صورت، یک ماتریس $m \times m$ برای اندازه‌گیری مقادیر کوواریانس نویز اندازه‌گیری هر یک از m لینک می‌باشد. همچنین برای یک فیلتر که n نرخ جریان قبلی را در نظر می‌گیرد، یک ماتریس $m \times n$ تعریف می‌شود، که دارای m سطر بهره خروجی هر یک با طول n ، برای نگهداری همه n نرخ جریان ترافیک قبلی است. بنابراین 'm' نشان‌دهنده تعداد لینک‌هایی است که در شبکه در نظر گرفته شده است، و 'n' نشان‌دهنده تعداد جریان‌های قبلی جریان ترافیک مورد بررسی است.

۲-۴- طراحی معیارهای خروجی

برای بررسی نتایج و عملکرد سیستم خود، بخشی از مدل را برای اندازه‌گیری دو مقدار زیر طراحی شده است:

• درصد خطای میانگین مربع ریشه ($RMSE$)

• بازده یا ($efficiency$)

برای این منظور ابتدا درصد خطا طبق رابطه زیر محاسبه می‌شود:

(۲۰)

$$\%error = \frac{actual\ traffic - predicted\ traffic}{actual\ traffic}$$

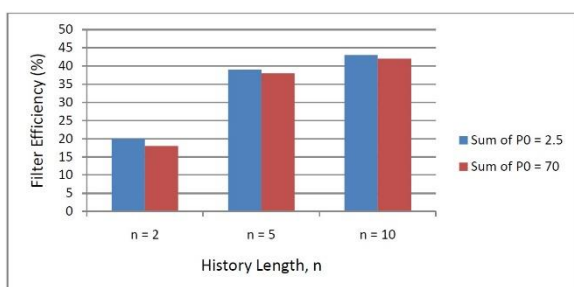
سیس میانگین مربع ریشه این خطا در مدل محاسبه می‌شود. برای محاسبه بازده فیلتر:

یک درصد خطای RMS معتبر ($rms_acceptable$)، در سیستم در نظر گرفته شد. که این مقدار نشان دهنده مقدار خطای پیش‌بینی قابل‌پذیرش کاربر است. و در هر تکرار فیلتر، درصد خطای RMS ، با عنوان ($rms_current$)، محاسبه می‌-

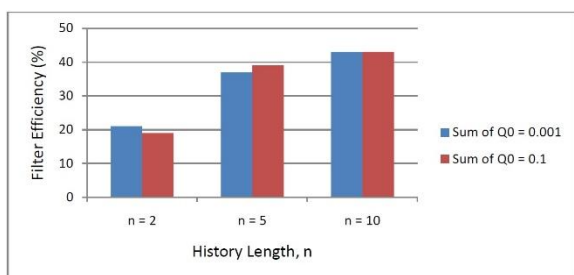
بر اساس نتایج فوق، پیش بینی با بازده بهتر در شرایط زیر حاصل می‌شود:

برای مقادیر بزرگتر n (و نه خیلی بزرگ، که اگر n خیلی بزرگ باشد، اطلاعات یکسان ممکن است تحریف شوند)، و در مقادیر متوسط n برای Q_0 بزرگتر و R کمتر. مقدار P_0 تاثیر کوچک یا هیچ اثری بر کارایی فیلتر ندارند.

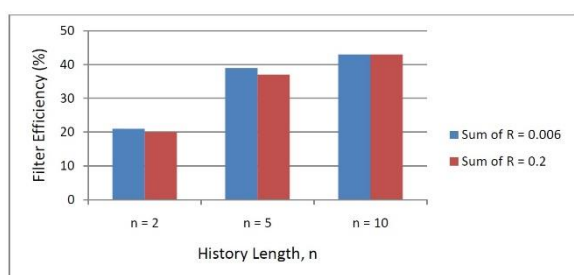
شکل‌های ۶-۱ تا ۶-۳ یک نمایش گرافیکی از نکات جدول بالا را نشان می‌دهد:



شکل ۶،۱: اثر n و P_0 بر کارایی فیلتر برای مجموعه داده با همبستگی قوی



شکل ۶،۲: تأثیر n و Q_0 بر کارایی فیلتر برای مجموعه داده با همبستگی قوی



شکل ۶،۳: اثر n و R در کارایی فیلتر برای مجموعه داده با همبستگی قوی

۶-۲ نتایج داده‌های شبکه:

در اینجا پارامترهای فیلتر با استفاده از نتایجی که در بالا ذکر شد تنظیم می‌شوند تا حداکثر بازده قابل دسترسی برای انواع مختلف داده‌های ترافیک شبکه بدست‌آید. و در نهایت سیستم با داده‌های واقعی ترافیک شبکه تست شد. جدول ۶،۲ نتایج حاصل از این فرایند را نشان می‌دهد:

*ماتریس R با یک ماتریس $m*m$ با مقادیر 0.15 ،

* مقدار $P_{\bar{R}}$ یک ماتریس $n*n$ با مقادیر بسیار کوچک 0.009

*ماتریس نویز فرآیند Q با مقادیر دلخواه و کوچک برای حالت پایدار و مقادیر کمی بزرگتر برای سیستم با تغییرات ناگهان [۶]. برای درک نتایج حاصل از آزمایشات، درک تاثیر مقادیر مختلف پارامترهای فیلتر مفید است.

الف: تاریخچه اندازه یا n نشان‌دهنده تعداد مقادیر داده‌های قبلی است. فیلتر به منظور پیش‌بینی مقدار بعدی تلاش می‌کند تا از هر گونه اطلاعات موجود در تاریخ داده استفاده کند.

ب: کوواریانس پیشین یا P یک اندازه از خطای پیش‌بینی است. P_0 به طور کلی باید بر کارایی فیلتر کم اثر یا کلا بی‌تاثیر باشد. پ: کوواریانس نویز فرآیند یا Q نشان‌دهنده نویز معرفی شده به فرایند پیش‌بینی است. مقدار کم برای Q یک مدل خوب را ایجاد می‌کند. Q_0 پیش‌بینی اولیه برای Q است.

ت: کوواریانس نویز اندازه‌گیری یا R نشان‌دهنده نویز وارد شده به فرایند پیش‌بینی در اثر خطای اندازه‌گیری مقادیر واقعی است. برای داده‌های ترافیک شبکه، R معمولاً ثابت است.

۶- شیب‌سازی

برای دستیابی به پیش‌بینی دقیق ترافیک شبکه، ابتدا پارامترهای مختلف فیلتر با یک مجموعه داده آزمایشی تست شد. و این نتیجه کلی رسیدیم که چگونه پارامترهای مختلف بر راندمان فیلتر تأثیر می‌گذارد. با استفاده از این دانش، برای انواع داده‌های ترافیک شبکه، گزینه‌های مناسب پارامتر فیلتر را می‌توان انتخاب کرد تا بازده فیلتر را به حداکثر برسانند.

بنابراین، مجموعه دوم آزمایش بر روی سه مجموعه داده واقعی ترافیک شبکه با ویژگی‌های مختلف اجرا شد. داده‌های واقعی مورد استفاده در جدول ۳،۳ نشان داده شده است.

۶-۱ نتایج داده‌های آزمایشی:

جدول ۶،۱ نتایج حاصل از اجرای فیلتر با مجموعه داده زیر را نشان می‌دهد.

(Data Set : ($R_{xx} = 4.1209$, $\sigma = 1.14$))

Table of filter efficiencies	$P_0 = 2.5$	$P_0 = 70$	$Q_0 = 0.001$	$Q_0 = 0.1$	$R = 0.006$	$R = 0.2$
n = 2	20	18	21	19	21	20
n = 5	39	38	37	39	39	37
n = 10	43	42	43	43	43	43

جدول ۶،۱: خروجی سیستم برای مجموعه داده آزمایشی

این فیلتر قادر به پردازش اطلاعات با ویژگی‌های مختلف بود، اما برای داده‌های با همبستگی زیاد و با واریانس کمتر، مناسب است. شایان ذکر است که این فیلتر در برنامه‌های پیش‌بینی ترافیک کوتاه مدت بهترین عملکرد را دارد.

Network Traffic Data	Rxx	σ	n	P ₀	Q ₀	R	Filter Efficiency (%)
Data 1	14175.3	55.81	10	2.5	0.25	0.0006	85
Data 2	125713	100.74	15	2.5	0.025	0.015	86
Data 3	450832	97.08	13	2.5	0.0185553	0.015	88

جدول ۶،۲: جدول بارده فیلتر برای داده‌های ترافیک شبکه

مراجع

- [1] R. J. Manish, H. T. Hassn, "A Review of Network Traffic Analysis and Prediction Techniques" Jalgaon Computer Science (M.S), India, 2015
- [2] H. Taheri, R. B. Sallah, M. F. Ab Razak, K. KO, N. Badrul Anuar, "Cost Effective Network Flow Measurement for Software Defined Networks: A Distributed Controller Scenario" Digital Object Identifier 10.1109/ACCESS.2017.2789281, 2018.
- [3] H. Farhady, H. Y. Lee, A. Nakao "Software-Defined Networking: A survey" Computer Networks. 81, pp. 79-95, 2015
- [4] White paper, "Software-Defined Networking: The New Norm for Networks, Open Networking Foundation" April 13, 2012. Retrieved August 22, 2013.
- [5] Z. Guo, Y. Xu, Z. Duan, Luo.Wang, S. Hui, J. Chao, "Improving the performance of load balancing in software-defined networks through load variance-based synchronization" Computer Networks, 68: p. 95-109, 2014.
- [۶] ف.فغانی، ق. میرجلیلی، ف. ادیب‌نیا، "نخمن پهنای باند موجود در شبکه‌های کامپیوتری با استفاده از فیلتر کالمن" سیزدهمین کنفرانس ملی انجمن کامپیوتر، کیش، ایران، ۱۳۸۶
- [7] C.Y. Hsu, P. W. Tsai, H. Y. Chou, M. Y. Luo, C. S. Yang, "A Flow-based Method to Measure Traffic Statistics in Software Defined Network" Network Research Workshop of the Asia-Pacific Advanced Network, v. 38, pp. 19-22, 2014.
- [8] J. Banks, J. S. Carson II, B. L. Nelson, D. M. Nicol, *Discrete-Event System Simulation*. Prentice Hall, pp478-500, 2010.
- [9] M. He, A. Basta, A. Blenk, W. Kellerer "Modeling Flow Setup Time for Controller Placement in SDN: Evaluation for Dynamic Flows" in Proceedings of 12th CNSM. IEEE, 2016.
- [10] M. Laaraideh "Implementation of Kalman Filter with Python Language" IETR Labs, University of Rennes, 2008
- [11] Z. Shu, J. Wan, J. Lin, S. Wang, D. Li, S. Rho, C. Yang "Traffic engineering in software-defined networking Measurement and Management" Digital Object Identifier 10.1109/ACCESS, 2016.
- [12] S. V. Kumar "Traffic Flow Prediction using Kalman Filtering Technique" 10th Int. Sci. Conf. on Transportation Science and Technology Procedia Engineering, Transbaltica, pp. 582 – 587, 2017.
- [13] H. Mohammadi, H. Yao, G. R. Khademi, T. T. Nguyen, D. Simon, H. Richter "Extended Kalman filtering for state estimation of a Hill muscle model" IET Control Theory Appl. Vol. 12, Iss. 3, pp. 384-394, 2018.
- [14] H. Fang, N. Tian, Y. Wang, M. Zhou, M. A. Haile "Nonlinear Bayesian Estimation: From Kalman Filtering to a Broader Horizon" IEEE/CAA JOURNAL OF AUTOMATICA SINICA, VOL. 5, NO. 2, MARCH 2018.
- [15] H. V. Lint, T. Djukic "Applications of Kalman Filtering in Traffic Management and Control" INFORMS, ISBN 978-0-9843378-3-5, 2012.
- [16] I.F. Akyildiz, A. L. P. Wang, L. Min, C. Wu, "A roadmap for traffic engineering in software defined networks" Computer Networks, vol. 71, p. 1-3, 2014.
- [17] S. Agarwal, M. Kodialam, T.V. Lakshman, "Traffic Engineering in Software Defined Networks" Proceedings IEEE INFOCOM, 2013.
- [18] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker "NOX: towards an operating system for networks. SIGCOMM Comput" Commun. Rev, vol.38, pp. 105-110, 2008.
- [19] <http://www.swarthmore.edu/NatSci/echeeve1/Ref/Kalman/MatrixKalman.html>
- [20] POXcontroller, Available: <http://www.noxrepo.org/pox/about-pox>
- [21] Mininet, Available: <http://www.mininet.org>

بازده نشان داده شده در جدول ۶،۲ پس از تغییر n و Q_0 و R بر اساس دستور العمل‌های قسمت قبل بدست آمد. شایان ذکر است:

الف: مقادیر نشان داده شده برای Q_0 و R مقادیر منحصر به فرد برای بدست آمدن حداکثر بازده نیست.

ب: مقدار نهایی n کوچکترین مقدار n است که برای آن حداکثر بازده ذکر شده در جدول ۶،۲، به دست می‌آید.

شکل ۶،۴ نمودار بازده به دست آمده برای مجموعه داده‌های مختلف بعد از تنظیم n ، Q_0 و R است که در جدول ۶،۲ نشان داده شده است.



شکل ۶،۴: کارایی فیلتر برای نمونه‌های مختلف داده شبکه

۷- نتیجه‌گیری

فیلتر کالمن یک ابزار برآورد و پیش‌بینی بسیار قدرتمند و با سازگاری مناسب است. این مدل فیلتر، یک منبع ساده، راحت و ارزان برای پیش‌بینی برنامه‌های کاربردی در شبکه‌های کامپیوتری و مخابراتی را فراهم می‌آورد. چنین مدلی راه حلی برای ارائه تکنولوژی ارزان قیمت و ساده برای کمک به کاهش بار ترافیک می‌تواند باشند. در این مقاله، یک مدل فیلتر کالمن با موفقیت برای پیش‌بینی ترافیک SDN، توسعه یافت. رفتار ترافیکی با موفقیت رصد، و فیلتر پیش‌بینی ترافیک شبکه را به میزان قابل توجهی از دقت (نزدیک به ۹۰٪) انجام داد. مسلماً مدل کامل نیست. کار بسیار بیشتر برای اصلاح رفتار مدل ضروری است، تا آن را قادر به پیش‌بینی با درجه دقت بالاتر و با تحمل خطا کمتر سازد.

[22] OpenvSwitch. Available: <http://openvswitch.org/>