

کاربرد های کاربردی هوش مصنوعی در شبکه های مبتنی بر نرم افزار

شیلا برهانزهی^۱، ساناز شهریاری^۲، مریم اصلی^۳

۱- دانشجوی رشته مهندسی نرم افزار کامپیوتر، دانشگاه رسالت، زاهدان، ایران (نویسنده مسئول)

۲- دانشجوی رشته مهندسی نرم افزار کامپیوتر، دانشگاه رسالت، زاهدان، ایران

۳- مدرس رشته مهندسی نرم افزار کامپیوتر، دانشگاه رسالت، زاهدان، ایران

چکیده

مقدمه: مطالعه ارائه شده کاربردهای مربوط به یک الگوی شبکه هوشمند مصنوعی به نرم افزار را که به عنوان بخشی از تلاش های گذشته برای دادن قابلیت برنامه ریزی به شبکه های کامپیوتری بسته به تفکیک بین کنترل ها و همچنین هواپیماهای انتقال در نظر گرفته می شود، بررسی می کند. با توجه به روش شبکه تعریف شده نرم افزار، کنترل کننده مغز مرکزی را در مورد شبکه مشخص می کند که منجر به سطح انعطاف پذیری پیچیده و همچنین هوش شبکه می شود. روش ها: بسیاری از رویکردهای مبتنی بر هوش مصنوعی برای دستیابی به تعادل بار، امنیت شبکه و همچنین کاربردهای هوشمند شبکه در روش شبکه های تعریف شده نرم افزار استفاده شد. یافته ها: سیستم مبتنی بر شبکه عصبی برای جلوگیری از نفوذ نشان دهنده عملکرد خوب با نرخ مثبت کاذب اندک است. استفاده از رویکرد مبتنی بر یادگیری تقویتی در سیستم های پخش ویدئوی تطبیقی در مقایسه با مسیریابی کوتاه ترین مسیر و همچنین روش مبتنی بر حریص، نرخ افت فریم را به میزان ۷۹٪ و ۶۰٪ کاهش می دهد. نتیجه گیری: کار ارائه شده تلاش اولیه برای اعمال هوش مصنوعی در مفهوم شبکه تعریف شده نرم افزار است. علاوه بر این، رویکردهای هوشمند ترکیبی ممکن است برای دستیابی به رفتار بهبود یافته در شبکه های مبتنی بر SDN از اهمیت بالایی برخوردار باشند.

واژه های کلیدی: شبکه های مبتنی بر نرم افزار، جریان باز، هوش مصنوعی

۱. مقدمه

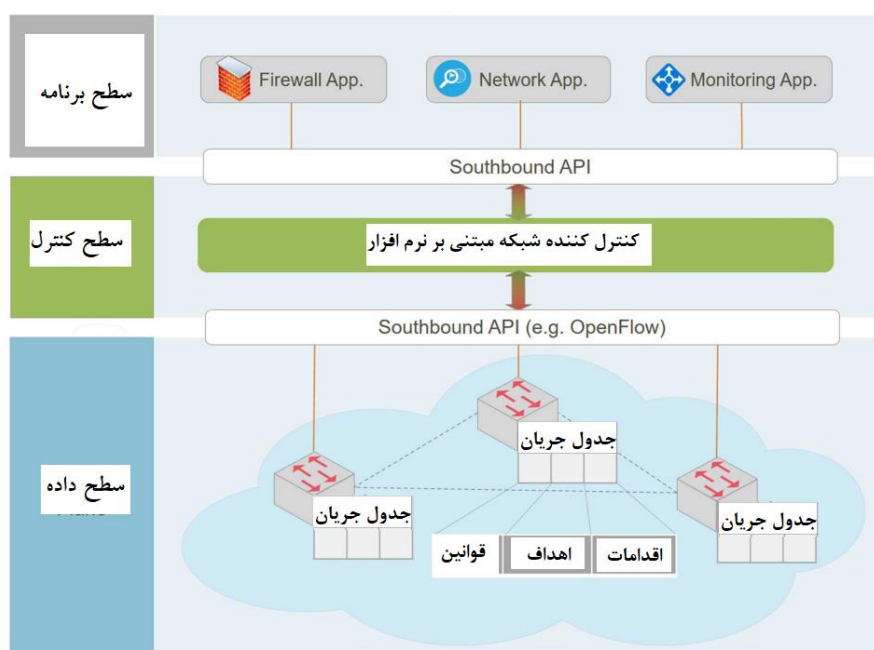
شبکه‌های مبتنی بر نرم‌افزار (SDN) مفهوم شبکه‌های قابل برنامه‌ریزی را با استفاده از یک مدیریت متمرکز منطقی اتخاذ می‌کند که یک راه‌حل ساده برای کارهای پیچیده مانند مهندسی ترافیک [۱]، بهینه‌سازی شبکه [۲] و ارکستراسیون [۳] را نشان می‌دهد. علاوه بر این، پرداختن به برنامه‌های کاربردی شبکه مدرن نیاز به معماری مقیاس پذیرتری دارد که باید بتواند خدمات قابل اعتماد و کافی را بر اساس یک نوع ترافیک خاص ارائه دهد [۴]. این را می‌توان با معماری SDN به دست آورد، که یک نمای کلی از وضعیت‌های شبکه را حفظ می‌کند و یک کنترل سطح جریان از لایه‌های زیرین را فراهم می‌کند [۴]. این ایده باعث تغییر چشمگیر در نحوه طراحی و مدیریت شبکه‌ها شد [۵، ۶]. علاوه بر این، معماری SDN اجازه می‌دهد تا اشخاص ثالث در طراحی و استقرار برنامه‌های کاربردی شبکه مدرن مشارکت داشته باشند [۶]. پروژه اتان [۷] پایه و اساس SDN امروزی را با ارائه یک الگوی شبکه جدید تشکیل داد که در آن یک کنترل کننده متمرکز برای مدیریت خط مشی سطح جریان و اهداف امنیتی در شبکه‌های سازمانی استفاده می‌شود. الگوی SDN صفحه کنترل را از صفحه داده جدا می‌کند، که منجر به دستیابی به رویکرد بسیار سریعتر و پویاتر در مقایسه با معماری شبکه معمولی می‌شود [۸]. صفحه کنترل را می‌توان به چندین شبکه مجازی تقسیم کرد که در آن هر یک سیاست متفاوتی را اجرا می‌کنند [۸]. در نتیجه، این پارادایم می‌تواند به عنوان ابزاری نگریننده شود که امکان پرداختن به مسائل مختلف در شبکه را از منظر دیگری فراهم می‌کند [۵] و همچنین می‌تواند برای برآوردن الزامات فناوری‌های جدید مانند اینترنت اشیا (IoT) و 5G [۹] مورد استفاده قرار گیرد. با این حال، پذیرش پارادایم SDN به شدت به موفقیت آن در دستیابی به راه حل مناسب برای مشکلاتی بستگی دارد که با پروتکل‌ها و معماری‌های شبکه سنتی قابل حل نیستند [۱۰]. برخی از شرکت‌های بزرگ مانند مایکروسافت و گوگل قبلاً استفاده از پارادایم SDN را برای مراکز داده خود آغاز کرده‌اند [۱۱، ۱۲]. از سوی دیگر، هوش مصنوعی (AI)، پتانسیل عظیمی را در نوآوری SDN نشان می‌دهد. مطالعه قبلی ما [۱۳] بر برجسته کردن اولین تلاش‌ها برای ادغام هوش مصنوعی در SDN متمرکز بود. با این حال، در این کار، ما یک نمای کلی از تلاش‌های تحقیقاتی در این زمینه برای به دست آوردن بینشی عمیق‌تر از نقش مهم هوش مصنوعی در پارادایم SDN ارائه می‌کنیم.

۲. بیان مسئله

۲.۱ شبکه‌های مبتنی بر نرم‌افزار (SDN)

همانطور که قبلاً ذکر شد، SDN نوآوری را با معرفی مفهوم کنترل قابل برنامه‌ریزی متمرکز صفحه داده، که توسعه سرویس‌ها و پروتکل‌های جدید شبکه را تسهیل می‌کند، ترویج می‌کند [۴]. معماری SDN بر اساس ایده جدایی بین صفحات کنترل و داده طراحی شده است (شکل ۱ را ببینید). اولین تلاش نقطه کنترل شبکه [۱۴] بود که از مفهوم جداسازی برای افزایش کنترل شبکه تلفن AT&T استفاده کرد، در حالی که مشارکت‌های اخیر مانند [۷] اتان و سانه [۱۵] همین مفهوم را برای شبکه‌های اترنت [۱۶] اعمال کردند. کاربردهای SDN در صفحه کاربردی معماری SDN قرار دارند که در آن رابط برنامه نویسی برنامه نویسی به شمال (API) جابجایی بین برنامه و سطوح کنترل را فراهم می‌کند [۸]، که امکان اجرای مجموعه‌ای از خدمات شبکه مانند مهندسی ترافیک، تشخیص نفوذ، کیفیت خدمات (QoS)، فایروال و برنامه‌های نظارتی [۴]. ارتباط بین صفحات کنترل و داده با استفاده از یک API به سمت جنوب مانند جداسازی عناصر انتقال و کنترل [۱۷]، پایگاه داده وی سوچ باز [۱۸]، ارسال نادیده گرفته پروتکل [۱۹]، حالت باز [۲۰]، جریان باز (جریان باز) [۲۱] ارائه می‌شود و اپفلکس [۲۲]، که تبادل پیام‌های کنترلی را با عناصر ارسال (به عنوان مثال سوئیچ‌های فعال جریان باز) امکان پذیر می‌کند. همانطور که در شکل ۱ نشان داده شده است، هر سوئیچ فعال جریان باز یک منطق تصمیم‌گیری مبتنی بر جریان را اتخاذ می‌کند که توسط

به اصطلاح کنترل کننده SDN تعیین می شود، که مسئول تهیه جداول ارسال هر سوئیچ است [۸]. یک سوئیچ معمولی دارای جریان باز دارای خط لوله ای از جداول جریان است که از ورودی های جریان تشکیل شده است که هر کدام دارای سه بخش است: (i) قوانین تطبیق که برای تطبیق بسته های ورودی استفاده می شود. (۲) شمارنده هایی که آمار جریان های منطبق را حفظ می کنند. و (iii) اقدامات یا دستورالعمل ها، که می توانند به صورت فعال یا واکنشی پیکربندی شوند تا بر اساس یک تطبیق اجرا شوند [۶، ۱۴]. عناصر ارسال (به عنوان مثال سوئیچ های فعال جریان باز) را می توان در نرم افزار یا سخت افزار پیاده سازی کرد. برخی از سوئیچ های نرم افزاری مانند وی سوئیچ باز پتانسیل زیادی برای ارائه راه حل برای مراکز داده و شبکه های مجازی دارند [۱۶]. از سوی دیگر، API های دیگر [۱۰، ۲۳] برای یک هدف خاص (مانند برنامه های VOIP و مسیریابی بین دامنه) پیشنهاد شده اند، به غیر از زبان های برنامه نویسی SDN مختلف مانند پروکرا [۲۴]، نت کر [۲۵] و فرنتریک [۲۶]، که API های سطح بالایی را ارائه می دهد که می تواند برای توسعه برنامه های مختلف SDN به شیوه ای انعطاف پذیرتر و کاربردی تر مورد استفاده قرار گیرد.



شکل ۱- ساختار پایه شبکه های مبتنی بر نرم افزار

۲.۲ پروتکل جریان باز

جریان باز به عنوان متداول ترین API جهت جنوب در SDN در نظر گرفته می شود که به طور مداوم توسط بنیاد شبکه باز در حال توسعه و استانداردسازی است [۶]. جریان باز یک لایه انتزاعی را فراهم می کند که کنترل کننده SDN را قادر می سازد تا به طور ایمن با عناصر ارسال فعال شده جریان باز ارتباط برقرار کند [۶]. جریان باز تبدیل به استاندارد واقعی برای API های جنوب به کار رفته در SDN ها شده است [۶] و بنابراین، در این مطالعه، ما عمدتاً بر روی SDN های مبتنی بر جریان باز تمرکز می کنیم. دستگاه های حمل و نقل مبتنی بر جریان باز برای همزیستی با دستگاه های اترنت معمولی توسعه یافته اند [۱۶]. سوئیچ های هیبریدی، از سوی دیگر، با قرار دادن هر دو درگاه OF و غیر OF، امکانات جدیدی را نشان می دهند [۶]. همانطور که قبلاً اشاره کردیم، مجموعه ای از پیام های کنترلی می تواند توسط کنترل کننده برای تهیه و به روز رسانی جداول جریان یک سوئیچ خاص ارسال شود. یک سوئیچ معمولی با قابلیت OF بسته های جدید را بر اساس جدول جریان خود کنترل

می کند. شایان ذکر است که کنترل دسترسی مبتنی بر هویت پروژه اتان [۷] به اولین مشخصات سوئیچ های OF تبدیل شد. پس از انتشار اولین مشخصات OF در بهار ۲۰۰۸، بسیاری از فروشندگان مانند اچ پی، ان ای سی، سیسکو و جونیپر توانستند اولین سوئیچ های سخت افزاری خود را با جریان باز بسازند، جایی که NOX [۲۷] تنها کنترل کننده موجود در آن دوره زمانی بود. [۲۸]. اخیراً، نسخه های مختلفی از پروتکل OF برای افزودن قابلیت های انعطاف پذیر و قابل اعتمادتر با گنجاندن جداول جریان چندگانه، توانایی های تطبیق/عمل پیشرفته، پورت های نوری، جداول گروه، جداول متر و جداول همگام سازی شده معرفی شده اند [۴، ۲۹]. جزئیات بیشتر در مورد نسخه های مختلف مشخصات OF را می توان در [۳۰] یافت. علاوه بر این، بسیاری از کنترل کننده های OF موجود هستند، مانند پاکس [۳۱]، بیکن [۳۲]، اپن دی لایت [۳۳]، فودلایت [۳۴] و ریو [۳۵]. تلاش های شبکه سازی تعریف شده نرم افزار برای افزایش سطح برنامه نویسی شبکه ها را می توان در ۳ مرحله دسته بندی کرد:

- شبکه های فعال: (از اواسط دهه ۹۰ تا اوایل این قرن) که منجر به اضافه شدن توابع شبکه قابل برنامه ریزی می شود. که در آن دستگاه های سوئیچینگ ممکن است رویه هایی را به منظور پردازش بسته ها انجام دهند.
- جداسازی صفحه کنترل و داده (از سال ۲۰۰۱ تا سال ۲۰۰۷) منجر به مهارت های جدیدی مانند کنترل یا پیش بینی رفتار مسیریابی می شود.

- رابط برنامه نویسی برنامه جریان باز (از سال ۲۰۰۷ تا سال ۲۰۱۰) که در ابتدا به عنوان رابط های جنوب بین کنترل و همچنین صفحه داده ای که جریان های شبکه L2L-4 را مدیریت می کند، استفاده شده است. با این حال، به منظور مدیریت جریان های L5L-7 برای ارائه قابلیت پشتیبانی از عملکرد مجازی سازی شبکه (NVF)، پروتکل جریان باز باید ۷ گسترش یابد. هر یک از جداول جریان شامل: (الف) فیلدهای سرصفحه، (ب) ضد و همچنین (ج) اقدامات. اقدامات قرار است در مواردی که تطبیق بسته ها رخ می دهد و شمارنده ها به روز می شوند استفاده می شوند.

۲.۳ هوش مصنوعی

هوش مصنوعی یک زمینه به سرعت در حال رشد است که شامل طیف گسترده ای از زمینه های فرعی از جمله نمایش دانش، استدلال، برنامه ریزی، تصمیم گیری، بهینه سازی، یادگیری ماشین (ML) و الگوریتم های فرا ابتکاری است. آزمون تورینگ [۳۶] تعریف کاملی از هوش ارائه می کند، که در آن رایانه باید به برخی از سوالات نوشته شده توسط یک بازجو پاسخ دهد. بر این اساس، اگر بازجو نتواند بگوید که پاسخ توسط یک انسان نوشته شده است یا یک ماشین، رایانه آزمایش را با موفقیت پشت سر می گذارد. برای قبولی در آزمون تورینگ، کامپیوتر باید قابلیت های پیشرفته ای مانند پردازش زبان طبیعی، نمایش دانش، استدلال خودکار، ام ال و چشم انداز کامپیوتر را در خود داشته باشد [۳۶]. تحقیقات هوش مصنوعی بعداً در اواسط دهه ۱۹۵۰ آغاز شد، جایی که یک کارگاه تابستانی توسط مارتین مینسکی و کلود شانون در کالج دارتموث سازماندهی شد و منجر به تولد رشته هوش مصنوعی شد [۳۷]. با این حال، اولین مشارکت در سال ۱۹۴۳ توسط مک کالوچ و پیتس انجام شد، زمانی که آنها اولین مدل را برای شبکه های عصبی مصنوعی پیشنهاد کردند که در آن هر نورون دارای یک خروجی باینری (۱-، ۱+) با تابع فعال سازی علامت است [۳۷]. پذیرش رویکردهای هوش مصنوعی به لطف کمک های کلیدی که منجر به ظهور زمینه های فرعی جدیدی مانند سیستم های خبره، منطق فازی (FL) و محاسبات تکاملی (EC) شد، افزایش یافت. تلاش های بیشتر با اصلاح روش های موجود و پیشنهاد رویکردهای هوشمند ترکیبی جدید، به تحقیقات در زمینه هوش مصنوعی دامن زده است. ام ال، الگوریتم های فرا ابتکاری و سیستم های استنتاج فازی به طور گسترده در پارادایم SDN استفاده می شوند، بنابراین ما مقدمه ای در رابطه با این رویکردها ارائه می کنیم.

۲.۳.۱ یادگیری ماشین

یک ماشین هوشمند از تجربه می آموزد (یعنی از داده های موجود در محیط خود می آموزد) و از آن برای بهبود عملکرد کلی استفاده می کند [۳۶، ۳۷]. در این زمینه روش های یادگیری به چهار گروه تقسیم می شوند.

۲,۴ هوش مصنوعی در شبکه های مبتنی بر نرم افزار

در حال حاضر، هوش مصنوعی و رویکردهای محاسباتی نرم در اکثر سیستم های اخیر مانند حمل و نقل هوشمند از اهمیت بالایی برخوردار هستند. این امکان را برای بهبود عملکرد در مورد شبکه های کامپیوتری موجود فراهم می کند. ادغام مفاهیم انتزاع در پارادایم SDN و همچنین روش های AI می تواند منجر به رفتار تطبیقی اضافی در مورد عناصر شبکه شود. علاوه بر این، قرار است رویکردهای جدیدی برای مقابله با مسائل متعارف شبکه و شبکه های تعریف شده نرم افزاری ارائه دهد. این بخش تلاش های فعلی را در چنین زمینه ای ارائه خواهد کرد [۳۸].

۲,۴,۱ موازنه بار و مسیریابی جریان

عملکرد تعادل بار را می توان به عنوان یک ضرورت برای به حداقل رساندن تأخیر و همچنین به حداکثر رساندن توان عملیاتی در شبکه های رایانه ای در نظر گرفت که از روش های متعدد مسیریابی پشتیبانی می کند. همچنین، تعادل بار را می توان به عنوان رویکرد دفاعی در برابر انواع خاصی از حملات شبکه مانند DDOS ۲۹,۳۰ تعریف کرد. انتزاع در SDN مزایای قابل توجهی را ارائه می دهد که نمای کلی و شناسایی توپولوژی مربوط به شبکه است. In ۳۱ شبکه عصبی انتشار برگشتی (BPNN) برای دستیابی به تعادل بار پویا در زمان واقعی و همچنین تأخیر ۱۹,۳ درصد در مقایسه با DLB و همچنین رویکردهای رند رابین ساکن کاهش یافته است. بردار ورودی برای NN شامل اطلاعات مسیر است که عبارتند از: الف. نسبت از دست دادن بسته، ب. نسبت استفاده از پهنای باند، ج. جهش انتقال، د. تأخیر انتقال مطالعه ای که توسط ۳۲ نفر انجام شد، روش مبتنی بر شبکه عصبی پس انتشار را برای تعادل بار در مراکز داده پیشنهاد کرد. BPNN در وی سوچ باز به صورت داخلی در رویکردی استفاده شد که زمان مصرف شده برای ارسال تصمیمات مسیریابی از کنترل کننده به وی سوچ باز را کاهش می دهد [۳۹].

۲,۵ کاربردهای شبکه هوشمند

ادغام هوش مصنوعی با SDN به ساخت برنامه های شبکه پیشرفته تر کمک می کند. مطالعه انجام شده توسط ۵۱ روش پیشنهادی از یک یادگیری تقویتی برای جریان ویدئوی تطبیقی در مفهوم شبکه تعریف شده نرم افزاری را پیشنهاد کرد. کنترل کننده به صورت دوره ای تصمیم گیرنده ای را که زمان انتخاب مسیر جدید و زمانی را که سرور تقاضای تغییر کیفیت ویدیو را درخواست می کند را تعیین می کند. فرآیند تصمیم گیری مارکوف برای مدل سازی اقدامات مرتبط با تصمیم گیری مورد استفاده قرار گرفت. رویکرد یادگیری Q در شرایط پاداش های ناشناخته برای حرکت بین حالت فعلی و بعدی استفاده می شود. درصد مربوط به از دست دادن بسته و همچنین تعداد تغییرات کیفیت نشان دهنده پارامترهای مهم اصلی برای تعریف پاداش است [۴۰].

۲,۶ سیستم های استنتاج فازی در شبکه های مبتنی بر نرم افزار

سیستم های استنتاج فازی نیز به طور گسترده در پارادایم SDN استفاده شد. پژوهش اصلی بر معرفی پروتکل های جدید، تشخیص نفوذ، انتخاب طرح های بهینه استقرار شبکه و مهندسی ترافیک متمرکز بود. عبدالملکی و همکاران یک پروتکل کشف توپولوژی برای شبکه های حسگر بی سیم تعریف شده با نرم افزار پیشنهاد کرد. پروتکل پیشنهادی آنها در مقایسه با راه حل SDN برای شبکه های حسگر بی سیم عملکرد بهتری را نشان داد [۴۱]. دنگ وان و تائو رویکردی برای جلوگیری از حملات DDOS بر اساس روش استنتاج فازی ساجنو ارائه کردند. آنها یک ترافیک شبکه واقعی را برای نتیجه گیری شاخص ها و آستانه های DDOS تجزیه و تحلیل کردند. ورودی ماژول فازی دو پارامتر زیر بود: نرخ بسته هایی که زمان های بین ورود در محدوده (۰-۲۰، اینچ) و نرخ جریان هایی که فقط یک بسته در هر جریان دارند. نتایج تجربی توانایی شناسایی و فیلتر کردن ۹۷ را نشان داد. علاوه بر این، نرخ مثبت کاذب نسبتاً پایینی (حدود ۵٪) به دست آورد و کاهش ورودی های جریان را ۵۰٪ در طول زمان حمله حفظ کرد [۴۲].

۳. نتیجه گیری

کار ارائه شده خلاصه ای از یکپارچگی بین رویکردهای هوش مصنوعی و پارادایم SDN را ارائه می دهد، همچنین معماری اصلی SDN و نقش مهم پروتکل جریان باز در آن را تعریف می کند. پس از آن، خلاصه ای از مشارکتهای فعلی برای ارائه رفتار پیشرفته تر شبکه در روش SDN ارائه کرد. شبکه های عصبی در برنامه های مختلفی مانند امنیت شبکه و متعادل سازی بار مورد استفاده قرار گرفتند. استفاده از رویکردهای هوش مصنوعی در سیستم های آگاه از امنیت SDN نشان دهنده کاهش نرخ تشخیص مثبت کاذب است. همچنین نتایج مربوط به سیستم پخش ویدئوی تطبیقی حاکی از کاهش نرخ افت فریم است. بنابراین، مطالعات در این فایل به سرعت در حال رشد هستند، همچنین روش هوشمند ترکیبی ممکن است پیشرفتهای بیشتری را در ناحیه مربوط به شبکه های مبتنی بر SDN به ارمغان بیاورد. در این مقاله، ما یک نمای کلی پیشرفته از تلاش های تحقیقاتی انجام شده برای استفاده از تکنیک های هوش مصنوعی در پارادایم SDN ارائه کردیم. در مقایسه با مقاله اخیر ما، این مطالعه افزایش پذیرش تکنیک های مختلف هوش مصنوعی برای حل طیف گسترده ای از مشکلات شبکه و رسیدگی به چالش های جدید معرفی شده توسط پارادایم SDN را نشان داد. این مطالعه نشان داد که زبان برنامه نویسی ML، فراابتکاری و FS ها رایج ترین رویکردهای AI در SDN هستند. علاوه بر این، نشان داده است که پیشرفتهای اخیر در تکنیک های ML مانند رویکردهای یادگیری عمیق و هوش مصنوعی ترکیبی می تواند نتایج بهتری را در مقایسه با رویکردهای سنتی ML ارائه دهد. به طور کلی، ثابت شده است که رویکردهای هوش مصنوعی ابزارهای بسیار مفیدی در SDN ها هستند. با این حال، تلاش های بیشتری برای مطالعه استحکام رویکردهای هوش مصنوعی تحت تنظیمات خصمانه نیز باید در نظر گرفته شود.

منابع:

- [1] Open Networking Foundation. Software-defined networking: The new norm for networks. Available from: <https://www.opennetworking.org/images/stories/downloads/sdnresources/white-papers/wp-sdn-newnorm.pdf>
- [2] Astuto BN, Mendonca M, Nguyen XN, Obraczka K, Turetti T. A survey of software-defined networking: past, present, and future of programmable networks. IEEE Communications Surveys and Tutorials. 2014; 16(3):1617–34.
- [3] Bakshi K. Considerations for Software Defined Networking (SDN): Approaches and use cases. IEEE Aerospace Conference, Big Sky; MT. 2013 Mar. p. 1–9.
- [4] Shinde MB, Tamhankar SG. Review: software defined networking and OpenFlow. International Journal of Scientific Research in Network Security and Communication. 2013Jun; 1(2):18–20.
- [5] Feamster N, Zegura E, Rexford J. The road to SDN: An intellectual history of programmable networks. ACM SIGCOMM Computer Communication Review archive. 2014; 44(2):87–98.
- [6] Nunes, B.A.A., Mendonca, M., Nguyen, X.N., et al.: ‘A survey of softwaredefined networking: past, present, and future of programmable networks’, IEEE Commun. Surv. Tutorials, 2014, 16, (3), pp. 1617–1634
- [7] Casado, M., Freedman, M.J., Pettit, J., et al.: ‘Ethane: taking control of the enterprise’, ACM SIGCOMM Comput. Commun. Rev., 2007, 37, (4), pp. 1–12

- [8] Jain, R., Paul, S.: 'Network virtualization and software defined networking for cloud computing: a survey', IEEE Commun. Mag., 2013, 51, (11), pp. 24–31
- [9] Bera, S., Misra, S., Vasilakos, A.V.: 'Software-defined networking for internet of things: a survey', IEEE Internet Things J., 2017, 4, (6), pp. 1994–2008
- [10] Lin, P., Bi, J., Wolff, S., et al.: 'A west-east bridge based SDN inter-domain testbed', IEEE Commun. Mag., 2015, 53, (2), pp. 190–197
- [11] Jain, S., Kumar, A., Mandal, S., et al.: 'B4: experience with a globally deployed software defined WAN', ACM SIGCOMM Comput. Commun. Rev., 2013, 43, (4), pp. 3–14
- [12] Hong, C., Kandula, S., Mahajan, R., et al.: 'Achieving high utilization with software-driven WAN'. Proc. of ACM SIGCOMM Conf., Hong Kong, China, August 2013, pp. 15–26
- [13] Latah, M., Toker, L.: 'Application of artificial intelligence to software defined networking: a survey', Indian J. Sci. Technol., 2016, 9, (44), pp. 1–7
- [14] Sheinbein, D., Weber, R.P.: 'Stored program controlled network: 800 service using spc network capability', Bell Labs Tech. J., 1982, 61, (7), pp. 1737–1744
- [15] Casado, M., Garfinkel, T., Akella, A., et al.: 'SANE: A protection architecture for enterprise networks'. Proc. 15th Conf. USENIX Security Symp, Vancouver, Canada, July 2006, pp. 1–15
- [16] Kreutz, D., Ramos, F.M., Verissimo, P.E., et al.: 'Software-defined networking: a comprehensive survey', Proc. IEEE, 2015, 103, (1), pp. 14–76
- [17] Internet Engineering Task Force (IETF): 'Forwarding and control element separation (ForCES) protocol specification [internet]'. 2010. Available at <http://www.ietf.org/rfc/rfc5810.txt>, accessed 5 December 2017
- [18] IETF: 'The open vswitch database management protocol [internet]'. 2013. Available at <https://tools.ietf.org/html/rfc7047>, accessed 5 December 2017
- [19] Song, H.: 'Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane'. Proc. 2nd ACM SIGCOMM Workshop Hot Topics Software Defined Networking, Hong Kong, China, August 2013, pp. 127–132
- [20] Bianchi, G., Bonola, M., Capone, A., et al.: 'Openstate: programming platform-independent stateful openflow applications inside the switch', ACM SIGCOMM Comput. Commun. Rev., 2014, 44, (2), pp. 44–51
- [21] McKeown, N., Anderson, T., Balakrishnan, H., et al.: 'Openflow: enabling innovation in campus networks', ACM SIGCOMM Comput. Commun. Rev., 2008, 38, (2), pp. 69–74
- [22] IETF: 'Opflex control protocol'. [Internet]. 2014. Available at <http://tools.ietf.org/html/draft-smith-opflex-00>, accessed 16 October 2017
- [23] Ferguson, A.D., Guha, A., Liang, C., et al.: 'Participatory networking: an API for application control of SDNs', ACM SIGCOMM Comput. Commun. Rev., 2013, 43, (4), pp. 327–338
- [24] Voellmy, A., Kim, H., Feamster, N.: 'Procera: a language for high-level reactive network control'. Proc. of the first workshop on Hot Topics in Software Defined Networks, Helsinki, Finland, August 2012, pp. 43–48
- [25] Monsanto, C., Foster, N., Harrison, R., et al.: 'A compiler and run-time system for network programming languages', ACM SIGPLAN Not., 2012, 47, pp. 217–230

- [26] Foster, N., Harrison, R., Freedman, M.J., et al.: 'Frenetic: a network programming language', ACM SIGPLAN Not., 2011, 46, (9), pp. 279–291
- [27] NOX - An OpenFlow Controller [Internet]. 2008. Available at: [http:// www.noxrepo.org](http://www.noxrepo.org), accessed: 10 December 2017
- [28] Kobayashi, M., Seetharaman, S., Parulkar, G., et al.: 'Maturing of OpenFlow and software-defined networking through deployments', Comput. Netw., 2014, 61, pp. 151–175
- [29] Dargahi, T., Caponi, A., Ambrosin, M., et al.: 'A survey on the security of stateful SDN data planes. IEEE Commun. Surv. Tutorials, 2017, 19, (3), pp. 1701–1725
- [30] ONF. OpenFlow Specification [Internet]. 2009. Available at: [https:// www.opennetworking.org/software-defined-standards/specifications/](https://www.opennetworking.org/software-defined-standards/specifications/) accessed 12 December 2017
- [31] POX Wiki [Internet]. 2011. Available at: <https://openflow.stanford.edu/display/ONL/POX+Wiki> accessed 20 December 2017
- [32] Beacon [Internet]. 2010. Available at: [https://openflow.stanford.edu/display/ Beacon/Home](https://openflow.stanford.edu/display/Beacon/Home), accessed 20 December 2017
- [33] OpenDayLight [Internet]. 2014. Available at <http://www.opendaylight.org/>, accessed 20 December 2017
- [34] Floodlight [Internet]. 2012. Available at [http://www.projectfloodlight.org/ floodlight/](http://www.projectfloodlight.org/floodlight/), accessed 12 December 2017
- [35] Ryu [Internet]. 2014. Available at <http://osrg.github.com/ryu/>, accessed 12 December 2017
- [36] Russell, S., Norvig, P.: 'Artificial intelligence (a modern approach)' (Prentice Hall, New Jersey, 1995, 3rd edn.), 1152p
- [37] Negnevitsky, M.: 'Artificial intelligence - a guide to intelligent systems' (Addison-Wesley, Essex, 2005, 2nd edn.), 415p
- [38] Nguyen, T.T., Armitage, G.: 'A survey of techniques for internet traffic classification using machine learning'. IEEE Commun. Surv. Tutor., 2008, 10, (4), pp. 56–76
- [39] Zhang, G.P.: 'Neural networks for classification: a survey', IEEE Trans. Syst. Man Cybern. C, Appl. Rev., 2000, 30, (4), pp. 451–462
- [40] Rokach, L.: 'Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography', Comput. Stat. Data Anal., 2008, 53, (12), pp. 4046–4072
- [41] LeCun, Y., Bengio, Y., Hinton, G.: 'Deep learning', Nature, 2016, 521, (7553), pp. 436–444
- [42] Deng, L.: 'A tutorial survey of architectures, algorithms, and applications for deep learning', APSIPA Trans. Signal Inf. Process., 2014, 3, (e2), pp. 1–19