مطالعه طبقه بندی ترافیک شبکه مبتنی بر الگوریتم‌های یادگیری ماشین

محمدرضا صمدزاده، موسسه آموزش عالی الکترونیکی ایرانیان - ایمیل : samadzadehm@gmail.com [1]

نجمه فرجی پور - استادیار موسسه آموزش عالی الکترونیکی ایرانیان - ایمیل : n.farajipour@iranian.ac.ir [2]

چکیده

شبکه تعریف شده نرم افزار یک الگوی انقلابی است که همراه با سایر فناوری‌های شبکه در روند نسل بعدی در حال رشد است و یک معماری برای تامین نیازهای شبکه مدرن با جداسازی داده‌ها و کنترل ترافیک است که مدیریت جریان متمرکز و قابلیت برنامه‌ریزی شبکه را امکان‌پذیر می‌کند. یکی از ایرادات معماری شبکه تعریف شده نرم افزار، که در شبکه‌های سنتی نیز وجود دارد، مربوط به آسیب‌پذیری بالای آن در برابر حملات انکار سرویس توزیع شده و سایر موارد مشابه به آن می‌باشد. از طرفی برای مقابله با این حملات از سیستم‌های تشخیص ناهنجاری استفاده می‌شود. جمع‌آوری داده‌ها و اطلاعات مربوط به شبکه، استخراج ویژگی‌های مؤثر و انتخاب بهترین مدل برای تشخیص ناهنجاری از جمله چالش‌های مربوط به طراحی این سیستم‌ها می‌باشد. در این پژوهش روش‌ها و انواع الگوریتم‌های یادگیری ماشین معرفی شده و به بررسی مفاهیم و چگونگی استفاده از یادگیری ماشین در طبقه‌بندی ترافیک شبکه پرداخته می‌شود.

واژه‌های کلیدی: یادگیری ماشین، ترافیک شبکه ، حملات کامپیوتری.

[1] موسسه آموزش عالی الکترونیکی ایرانیان

[2] موسسه آموزش عالی الکترونیکی ایرانیان

۱- مقدمه

شبکه‌های کامپیوتری سنتی از تعداد زیادی تجهیزات روانه سازی[1]، یعنی مسیریاب‌ها و/یا سوئیچ‌ها تشکیل شده‌اند که توسط پروتکل‌های زیادی اداره می‌شوند و طیف وسیعی از برنامه‌ها را اجرا می‌کنند. وجود این ناهمگونی در زیرساخت، مدیریت شبکه و بهینه‌سازی عملکرد را پیچیده‌تر می‌کند. شبکه‌های سنتی سیستم‌های توزیع‌شده‌ای هستند که در آن هر تجهیز روانه سازی یک نمای محلی را در کل شبکه حفظ می‌کند. بنابراین، استفاده از تکنیک‌های یادگیری ماشین در سیستمی که عناصر آن دارای دید محدودی هستند، یک چالش بزرگ دیگر است[۱].به وجود آمدن کاربردهای فراوان شبکه موبایل، شبکه‌های اجتماعی، سرویس‌های ابری و داده‌های بزرگ در فناوری اطلاعات و ارتباطات باعث ایجاد چالش‌های جدیدی مانند امکان دسترسی مستمر در شرایط مکانی و زمانی مختلف، پهنای باند بالا و مدیریت پویای شبکه در رابطه با آینده شبکه‌های کامپیوتری و اینترنت شده است. با توجه به اندازه، ناهمگون بودن و پیچیدگی‌های شبکه‌های فعلی و احتمالاً شبکه‌های آینده، روش‌های متداول و قدیمی برای پیکربندی، بهینه‌سازی و عیب‌یابی ممکن است غیر موثر باشند و در برخی از موارد، حتی ممکن است باعث ایجاد خطا شوند. توسعه سریع اینترنت و دستگاه‌های ارتباطی ساختارهای شبکه بزرگ‌تر و پیچیده‌تری را ایجاد کرده است و هاب‌ها، روترها، سوئیچ‌ها و سایر تجهیزات بزرگ‌تری را تطبیق داده و توسعه می‌دهد. این پیچیدگی در شبکه‌ها باعث ایجاد سرریز حجم وسیعی از داده‌های ترافیک شده و به چالش‌های مدیریت شبکه و بهینه‌سازی ترافیک، ازجمله اندازه‌گیری و طبقه‌بندی ترافیک مسائل زیادی را اضافه کرده است. در این پژوهش روش‌ها و انواع الگوریتم‌های یادگیری ماشین معرفی شده و به بررسی مفاهیم و چگونگی استفاده از یادگیری ماشین در طبقه‌بندی ترافیک شبکه پرداخته می‌شود.

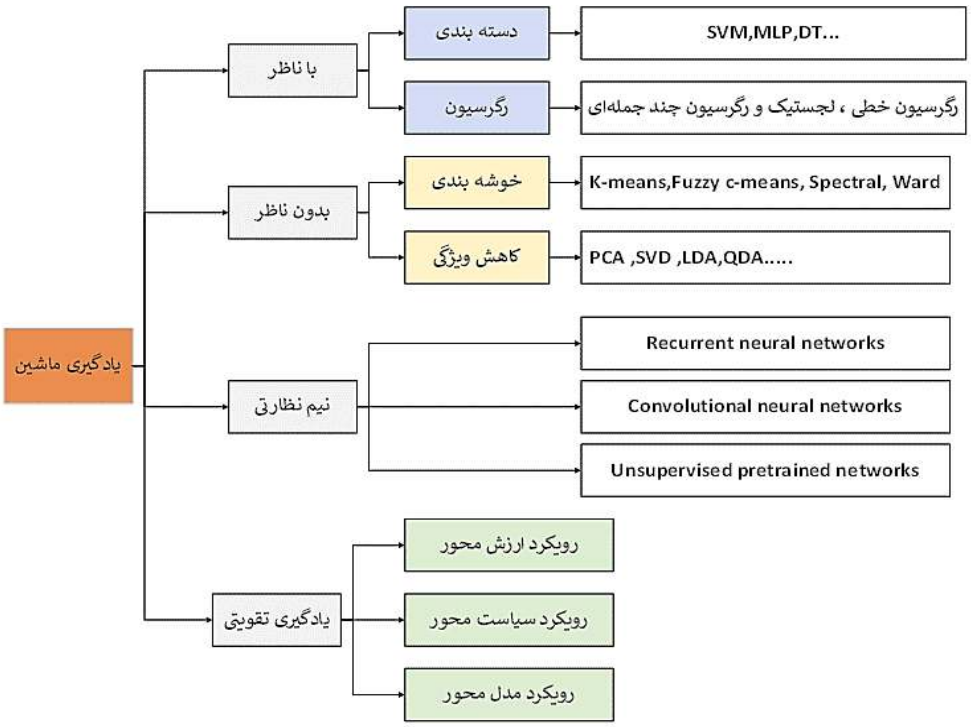
۲- یادگیری ماشین

یادگیری ماشین یک روش آنالیز داده است که با یادگیری داده‌ها، الگوهای درونی آن‌ها را شناسایی کرده و بر اساس اطلاعات جمع‌آوری‌شده تصمیم‌گیری می‌کند. این روش به‌طورکلی شامل مراحل پیش‌پردازش، آموزش و آزمایش می‌باشد. پیش‌پردازش شامل اقداماتی مانند آماده‌سازی داده‌ها، فیلتر کردن، انتساب و تنظیم برای اهداف مشخص می‌باشد. هنگامی‌که داده‌ها پیش‌پردازش شدند، روش‌های یادگیری ماشین برای آموزش داده‌ها اجرا می‌شوند. سپس سیستم بر اساس ورودی‌های دریافتی از مرحله آموزش تصمیم‌گیری می‌کند. الگوریتم‌های یادگیری ماشین را می‌توان به دو صورت مورد مطالعه قرار داد، یادگیری نظارت‌شده و یادگیری بدون نظارت. در یادگیری نظارت‌شده از داده‌های آموزشی برچسب‌گذاری شده استفاده می‌شود اما در یادگیری بدون نظارت از داده‌های آموزشی بدون برچسب استفاده می‌شود، این روش سعی دارد اطلاعات را از طریق دسته‌بندی بر اساس میزان شباهت در نقاط مشاهده استخراج کند. در ادامه مفاهیم بیشتر در مورد الگوریتم‌های یادگیری ماشین ارائه‌شده است و به تشریح الگوریتم‌ها و روش‌های مختلف یادگیری ماشین و یادگیری عمیق پرداخته شده است.

۳- دسته‌بندی الگوریتم‌های یادگیری ماشین

در این بخش به معرفی و شرح الگوریتم‌های یادگیری ماشین و مفاهیم مرتبط پرداخته می‌شود. الگوریتم‌های یادگیری ماشین به چهار دسته اصلی تقسیم می‌شوند که عبارت‌اند از: ۱- الگوریتم‌های با ناظر[2]، ۲- الگوریتم‌های بدون ناظر[3]، ۳- الگوریتم‌های نیمه نظارتی[4] و ۴- الگوریتم‌های یادگیری تقویتی[5][۲]. دسته‌بندی کلی الگوریتم‌های یادگیری ماشین در شکل (۱) نشان داده شده است.

[1] forwarding elements
[2] Supervised
[3] Unsupervised
[4] Semi-Supervised
[5] Reinforcement Learning



شکل ۱- دسته‌بندی کلی روش‌های یادگیری ماشین

۳-۱- یادگیری نظارتی

یادگیری نظارت‌شده، بر اساس برچسب‌های از پیش تعریف‌شده، اطلاعاتی از نمونه‌های جدید ارائه می‌کند. این مدل یادگیری ماشین توسط مجموعه‌ای از ورودی‌ها با خروجی‌های مشخص آموزش داده می‌شود. این نوع یادگیری به مدل اجازه می‌دهد تا ویژگی‌ها را بررسی کند و سپس روابطی را برای پیش‌بینی برچسب کلاس نمونه‌های دیده نشده ایجاد کند. از یادگیری نظارت‌شده برای ساخت سیستمی استفاده می‌شود که با استفاده از قوانین از پیش آموخته‌شده، خروجی را بر اساس یک ورودی پیش‌بینی می‌کند. یادگیری نظارتی شامل دو نوع اصلی است: طبقه‌بندی و رگرسیون. در طبقه‌بندی، مدل برچسب کلاس را پیش‌بینی می‌کند که این برچسب یک خروجی طبقه‌بندی‌شده از پیش تعریف‌شده می‌باشد. اما در رگرسیون یک خروجی پیوسته پیش‌بینی می‌شود. یادگیری نظارت‌شده شامل دو مرحله آموزش و آزمایش می‌باشد. در طول آموزش، مدلِ طبقه بند جهت بررسی مجموعه داده ارائه‌شده ساخته می‌شود. در طول آزمایش، مدل طبقه بند به‌طور خودکار کلاس‌های آموخته‌شده را به مجموعه داده آزمایشی حاوی نمونه‌های دیده نشده تخصیص می‌دهد. معروف‌ترین الگوریتم‌های یادگیری نظارت‌شده عبارت‌اند از جنگل تصادفی[1]، K-نزدیک‌ترین همسایه(KNN[2])، درخت تصمیم[3]، شبکه عصبی[4]، و ماشین بردار پشتیبان (SVM[5])[۳].

۳-۲- یادگیری غیرنظارتی

برخلاف یادگیری نظارتی، در اینجا داده‌های ورودی یا داده‌های آموزشی برچسب‌گذاری نشده است، لذا این نوع یادگیری سخت‌تر است. در یادگیری بدون نظارت، یک ورودی بدون برچسب به الگوریتم یادگیری ماشین ارائه می‌شود. بنابراین، خروجی یک نمونه نیز تعریف نشده

[1] Random Forest
[2] K-Nearest Neighbor
[3] Decision Tree
[4] Neural Network
[5] Support Vector MAchine

است. یادگیری بدون نظارت بدون راهنمایی اجرا می‌شود و هدف آن یافتن یک الگو یا ساختار در داده‌های ورودی برای گروه‌بندی آن‌ها بر اساس شباهت یا روابط آماری بین ویژگی‌ها است. در اینجا وظیفه ماشین جمع‌آوری اطلاعات با توجه به شباهت‌ها، الگوها و تفاوت‌ها بدون آموزش قبلی داده‌ها است؛ بنابراین، ماشین به یافتن ساختار پنهان در داده‌های بدون برچسب توسط الگوریتم محدود می‌شود. برای مثال، فرض شود یک تصویر وجود دارد که هم سگ‌ها و هم گربه‌ها را دارد که تابه‌حال دیده نشده است؛ بنابراین ماشین هیچ ایده‌ای در مورد ویژگی‌های سگ و گربه ندارد و نمی‌تواند آن را در سگ و گربه‌ها دسته‌بندی کند؛ اما می‌تواند آن‌ها را با توجه به شباهت‌ها، الگوها و تفاوت‌ها دسته‌بندی کند. در این حالت سیستم چیزی آموزش نمی‌بیند به این معنی که هیچ داده یا نمونه آموزشی از قبل وجود ندارد. یادگیری غیرنظارتی به دو دسته الگوریتم خوشه‌بندی و انجمن تقسیم شده است اگر اطلاعات خرید افراد در یک فروشگاه در نظر گرفته شود، خوشه‌بندی زمانی می‌باشد که هدف گروه‌بندی ذاتی مانند گروه‌بندی مشتریان با رفتار خرید موجود در داده‌ها باشد؛ اما انجمن مانند قانون یک رابطه است که در آن افراد می‌خواهند قوانینی را کشف کنند که بخش‌های بزرگی از داده‌ها را توصیف می‌کنند، مانند افرادی که X را می‌خرند نیز تمایل به خرید Y دارند[۳].

۳-۳- یادگیری نیمه نظارتی

در یادگیری نیمه نظارتی داده های آموزش حاوی داده برچسبگذاری شده و برچسبگذاری نشده هستند. هدف درواقع توسعه یک الگوریتم است که دسته های داده تست آینده را بهتر از الگوریتم قبلی (که تنها از داده های برچسبگذاری شده استفاده می‌کرد) پیش‌بینی کند. شیوه یادگیری انسان مشابه یادگیری نیمه نظارتی است.

۳-۴- یادگیری تقویتی

در این نوع یادگیری الگوریتم نتیجه اقدام خود را به وضعیت نگاشت می‌کند و یک پاداش یا جریمه به خاطر اقداماتی که در حل یک مسئله انجام داده است دریافت می‌کند. بعد از چند خطا و آزمایش بهترین سیاست را یاد می‌گیرد، یعنی توالی اقداماتی که پاداش نهایی را ماکزیمم می‌کند.

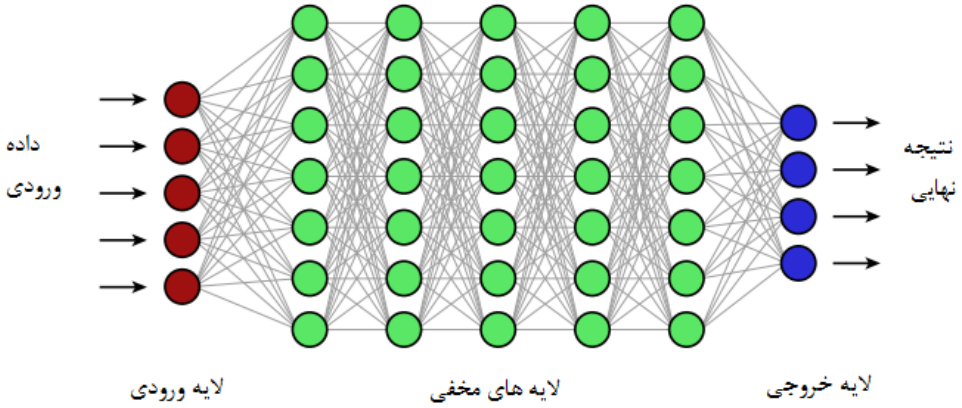
۳-۵- یادگیری عمیق

تکنیک‌های یادگیری ماشین نوع ضعیفی از هوش مصنوعی هستند و بنابراین کاملاً مستقل نیستند و به راهنمایی نیاز دارند، به‌عنوان‌مثال هنگام تنظیم ابرپارامترها[1]. برای فراتر رفتن از رویکردهای سنتی یادگیری ماشین، روش‌های یادگیری عمیق (DL[2]) توسعه یافتند که سعی می‌کنند برخی از جنبه‌های انسانی را با دقت بیشتری تقلید کنند. این کار به آن‌ها اجازه می‌دهد تا بدون نیاز به مداخله دستی برنامه‌نویس، در دقت و سرعت از سایر الگوریتم‌های یادگیری ماشین بهتر عمل کنند. یادگیری عمیق یک نوع یادگیری ماشین است که ورودی‌های خود را از طریق یک معماری شبکه عصبی مصنوعی با الهام از ساختار بیولوژیکی اجرا می‌کند. با گذشت زمان، مشخص شد که شبکه‌های عصبی به دلیل توانایی قوی خود در استخراج ویژگی‌ها و اطلاعات مربوطه در حجم زیادی از داده‌ها، ازنظر دقت و سرعت از بسیاری از الگوریتم‌های دیگر بهتر عمل می‌کند. یادگیری عمیق قادر به مدل‌سازی و پردازش روابط غیرخطی بسیار پیچیده می‌باشد و این مسئله آن را برای کاربرد در مسائل مختلف مناسب می‌سازد.

روش‌های یادگیری عمیق مختلفی وجود دارد، ازجمله شبکه‌های عصبی کانولوشن (CNN[3])، شبکه‌های عصبی بازگشتی (RNN[4])، شبکه‌های عصبی مصنوعی (ANN[1])، و شبکه‌های عصبی عمیق (DNN[2]). شبکه‌های عصبی حاوی نورون‌های مصنوعی هستند که در

[1] Hyperparameters
[2] Deep Learning
[3] convolutional neural networks
[4] Recurrent neural networks

چندین لایه چینش یافته‌اند به‌طوری‌که هر لایه فقط با لایه‌های قبل و بعد خود ارتباط برقرار می‌کند. ساختار یک شبکه عصبی عمیق در شکل۲ نشان داده شده است.



شکل۲- ساختار یک شبکه عصبی عمیق

۴- خودرمزگذار[3]

الگوریتم‌های یادگیری عمیق خودرمزگذار و شبکه‌های عصبی پیچشی از الگوریتم‌های موفق در حل بسیاری از مسائل داده‌کاوی و یادگیری ماشین می‌باشند. خودرمزگذار همانند شبکه عصبی پرسپترون چندلایه(MLP[4]) ولی به‌صورت یک شبکه بدون نظارت عمل می‌کند. در این شبکه ورودی ویژگی‌های موجود از داده‌ها هست اما الگوریتم به‌جای دسته‌بندی داده‌ها، ورودی را به یک فضای انتزاعی در لایه میانی نگاشت کرده و سپس در لایه خروجی داده‌های ورودی را از ویژگی‌های انتزاعی استخراج‌شده در لایه میانی بازسازی می‌کند. خودرمزگذار به‌صورت سلسله‌مراتبی به کاهش بُعد داده‌های ورودی می‌پردازد [۴]. این شبکه‌ها دارای یک‌لایه‌ی ورودی، یک‌لایه‌ی خروجی و یک‌لایه‌ی پنهان[5] هستند [۵].

۵- شبکه‌های عصبی پیچشی(CNN[6])

این شبکه شبیه به شبکه عصبی پرسپترون چندلایه است که شامل یک توالی از عملیات است. این عملیات معمولاً عبارت‌اند از: کانولوشن (پیچش) و تجمع (ادغام) است. شبکه‌های عصبی پیچشی دارای یک‌لایه‌ی ورودی، یک‌لایه‌ی خروجی و یک‌لایه‌ی کانولوشنی با چندین فیلتر با ابعاد متفاوت و به دنبال آن یک‌لایه‌ی pooling است.

۶- شبکه عصبی بازگشتی (RNN[7])

شبکه عصبی بازگشتی یک شبکه با حلقه[8] است که ورودی را در حافظه داخلی خود نگهداری می‌کند. درست مانند رفتار انسان، این روش اطلاعات فعلی و تجربیات قبلی که از طریق حلقه‌ها به دست آورده است را در طول تصمیم‌گیری لحاظ می‌کند. محبوب‌ترین شبکه عصبی

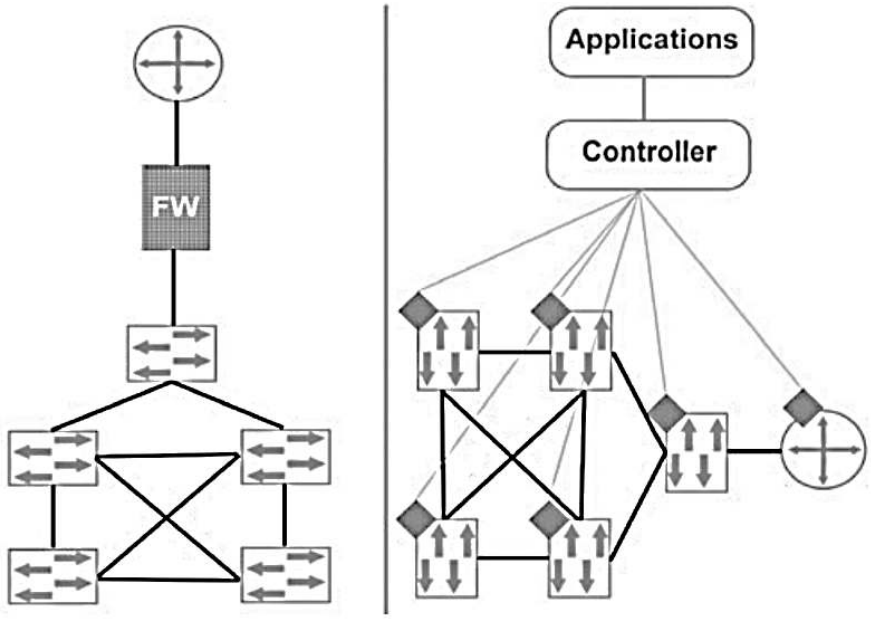
---

[1] artificial neural networks
[2] deep neural networks
[3] Auto-Encoder
[4] Multilayer Perceptron
[5] Hidden Layer
[6] Convolutional Neural Network
[7] Recurrent Neural Network
[8] loop

بازگشتی، حافظه طولانی کوتاه‌مدت (LSTM[1]) می‌باشد. این روش خطاها را از طریق لایه‌ها به عقب انتشار می‌دهد تا به روشی بازگشتی یاد بگیرد.

۷- معماری کلی شبکه نرم‌افزار محور

برخلاف شبکه‌های مرسوم IP (شکل ۳) که عملکردهای آن‌ها غیرمتمرکز است، SDN یک شبکه متمرکز است که دامین‌های شبکه‌های اتصالی را بین صفحه کنترل و صفحه داده و در یک زیرساخت یکسان ارائه می‌دهد. علاوه بر این، SDN امکان سازگاری با پروتکل‌ها و استانداردهای موجود (مانند IP، ARP، VLAN، اترنت و غیره) را فراهم می‌کند[۶].



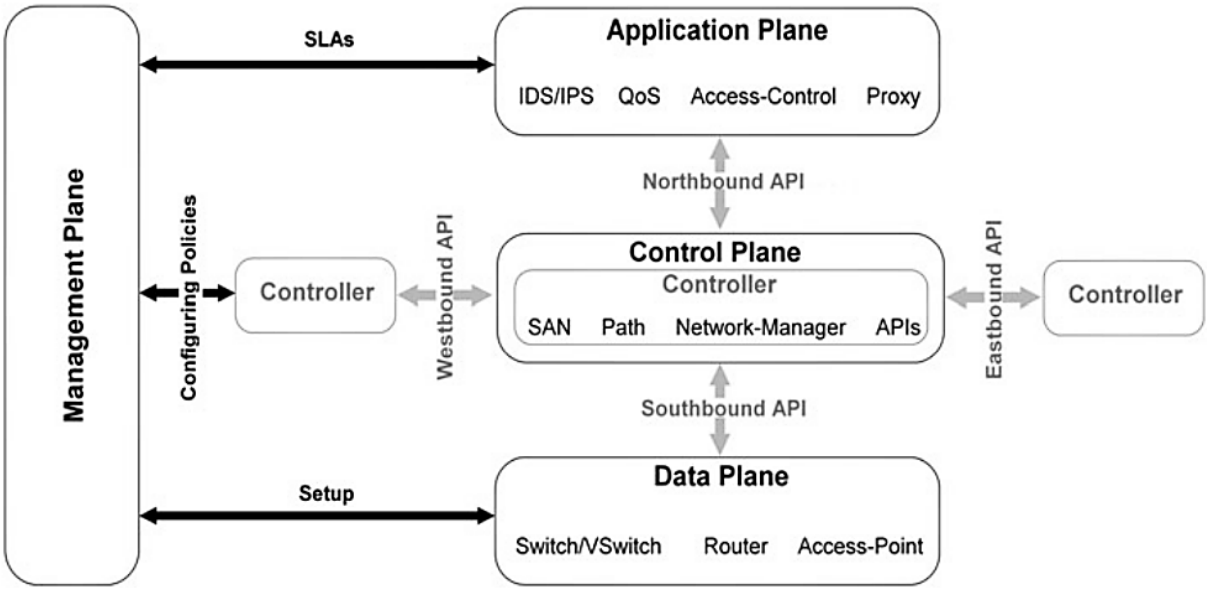
شکل۳- ساختار شبکه SDN در مقابل ساختار شبکه‌های سنتی[۶]

در شکل ۳، معماری هسته SDN به سه لایه تقسیم شده است. لایه بالایی معماری SDN لایه کاربرد[2] است که قوانین را تعریف می‌کند و خدمات مختلفی مانند فایروال، کنترل دسترسی، IDS/IPS، کیفیت خدمات، مسیریابی، سرویس پروکسی و متعادل‌کننده یا بالانسر[3] نظارت را ارائه می‌دهد. این لایه مسئول انتزاع مدیریت کنترل شبکه SDN از طریق واسط API شمالی (به‌عنوان‌مثال، OpenDaylight) است. لایه دوم تحت عنوان صفحه کنترل شناخته می‌شود که انتزاعی از توپولوژی شبکه است. کنترل‌گر اصلی‌ترین عضوی است که مسئول ایجاد جداول جریان و سیاست‌های مدیریت داده و همچنین انتزاع پیچیدگی شبکه و جمع‌آوری اطلاعات شبکه از طریق واسط API جنوبی و حفظ یک نمای به‌روز و جامع از شبکه است. ارتباطات واسط API جنوبی[4] می‌توانند در دو حالت گسترش یابند:

[1] Long Short-Term Memory
[2] application
[3] balancer
[4] southbound API

- ارتباطات درون باندی[1]: در این سناریو، ترافیک بین کنترل‌کننده و هر دستگاه شبکه باید از قوانین جریان مشخصی پیروی کند.
- ارتباطات خارج از باندی[2]: در این سناریو، ترافیک از قوانین جریان پیروی نمی‌کند. این نوع ارتباط نیازمند اجرای VLAN برای جداسازی جریان ترافیک از ارتباطات است، که به قوانین OpenFlow بستگی دارد.

همچنین واسط‌های API شرقی و غربی (به‌عنوان‌مثال HyperFlow) وجود دارد که چندین کنترل‌کننده را قادر می‌سازد تا اطلاعات کنترلی مربوط به جریان صفحه داده را مبادله کنند. ما می‌توانیم چندین کنترل‌کننده بر اساس زبان‌های برنامه‌نویسی مختلف (Python، C/C++، Java، Ruby، و غیره) و پلتفرم‌هایی مانند NOX ، Floodlight، Beacon، Maestro و Trema بیابیم. پایین‌ترین لایه که تحت عنوان لایه صفحه داده شناخته می‌شود، دستگاه‌های شبکه مانند سوئیچ‌های فیزیکی/مجازی، روترها و نقاط دسترسی[3] را فراهم می‌آورد و مسئول تمام فعالیت‌های مربوط به داده‌ها ازجمله ارسال روبه‌جلو[4]، قطعه‌قطعه کردن[5] و بازسازی مجدد[6] است.



شکل ۴- معماری کلی شبکه‌های نرم‌افزار محور[۶]

۸- مدل‌های مبتنی برابر شبکه‌های موجود

مدل‌های مبتنی برابر متعددی برای بهبود عملکرد شبکه و تسهیل مدیریت زیرساخت‌های شبکه مانند NaaS وجود دارد. به‌عنوان‌مثال، یک مدل SDN مبتنی بر OpenFlow را می‌توان برای پشتیبانی از NaaS، بهبود عملکرد و اعمال دید کلی از شبکه به کار برد. علاوه بر این، طیف گسترده‌ای از برنامه‌های کاربردی قابل‌اعتماد به کاربران ارائه می‌شود تا از عملکردهای شبکه پیشرفته بهره‌مند شوند. همچنین دو مدل مبتنی برابر دیگر نیز وجود دارد: مدل مجازی‌سازی شبکه و مدل تکاملی. مدل مجازی‌سازی شبکه را می‌توان برای قابلیت مقیاس‌پذیری در VLAN به SDN اعمال کرد. این رویکرد امکان ایجاد چندین نمونه شبکه مجازی را در یک زیرساخت فیزیکی واحد

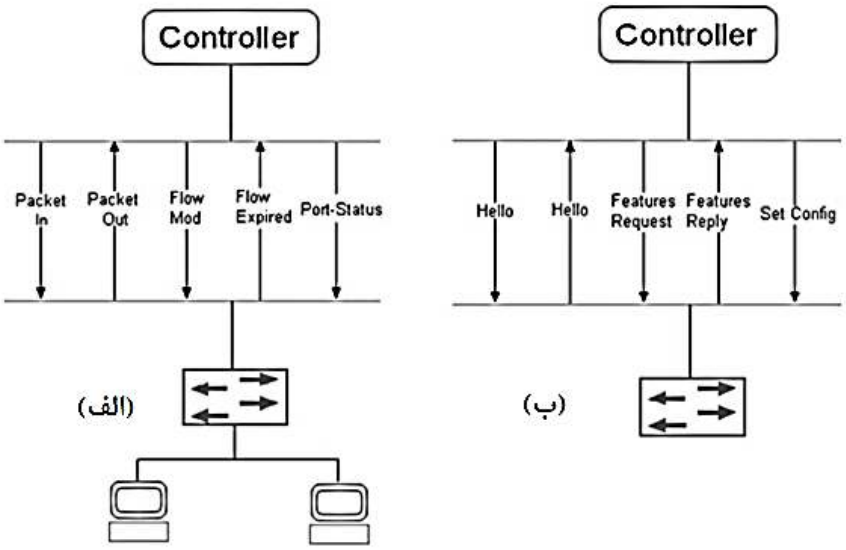
[1] in-band
[2] out-band
[3] access point
[4] forwarding
[5] fragmentation
[6] reassembly

فراهم می‌کند. این مدل برای حل مسائل مربوط به پویایی زیرشبکه‌ها و ایزوله سازی در محیط‌های مشترک معرفی شده است. هدف مدل تکاملی تقسیم شبکه به بخش‌های مجازی و تضمین کیفیت خدمات، تحمل خطا[1] و مدیریت پیکربندی و غیره می‌باشد.

۸-۱- پروتکل ارتباطات

متداول‌ترین واسط جنوبی در SDN به OpenFlow متکی است. خصوصیات OpenFlow پروتکلی را تعریف می‌کند که کنترل‌گر را قادر می‌سازد تا به سوئیچ‌ها و روترها فرمان دهد.

سه راه ارتباطی در پروتکل OpenFlow وجود دارد: ارتباطات کنترل‌کننده به سوئیچ، ناهمگام و متقارن. ارتباط کنترل‌کننده به سوئیچ مسئول پیکربندی دست تکانی[2]، سوئیچ و جدول جریان است. ارتباط ناهمگام[3] از سوئیچ سازگار با OpenFlow شروع می‌شود تا با ارسال پیامpacket-in ، پیام وضعیت پورت و پیام حذف جریان، کنترل‌کننده را مطلع کند (شکل ۵ الف). برای ارتباطات متقارن، هیچ محدودیتی در مورد اینکه چه کسی مبادلات را آغاز می‌کند وجود ندارد. نمونه‌هایی از این پیام‌ها Hello و Echo هستند که برای کمک به شناسایی مشکلات اتصال سوئیچ-کنترل‌کننده استفاده می‌شوند. پیام‌های دیگر عبارت‌اند از Feature Request/Reply، Set Config و غیره (شکل ۵ ب).



شکل۵- تبادلات OpenFlow

بسته به نیازهای امنیتی، ارتباط بین کنترل‌گر و سوئیچ می‌تواند از طریق امنیت لایه حمل‌ونقل (TLS) یا بدون مکانیزم رمزگذاری برقرار شود.

۹- آسیب‌پذیری‌های شبکه‌های نرم‌افزار محور

SDN یک طراحی شبکه جدیدی را ارائه می‌دهد که طیف گسترده‌ای از برنامه‌ها و خدمات شبکه را توانمند می‌سازد اما بااین‌حال نگرانی‌های امنیتی به یک قضیه بسیار مهمی تبدیل شده است، زیرا امنیت همچنان یک ویژگی داخلی در معماری SDN محسوب نمی‌شود. بر اساس فناوری‌های موجود (به‌عنوان‌مثال، روترها، سوئیچ‌ها، سرورها، برنامه‌ها و غیره)، SDN مسائل امنیتی مرتبط به خود را از

[1] fault tolerance
[2] handshake
[3] asynchronous

لایه فیزیکی تا لایه کاربرد مدل اتصال متقابل سیستم‌های باز[1] به ارث می‌برد. درحالی‌که SDN عناصر جدید (مانند کنترل‌کننده‌ها، اجزای مجازی‌سازی شده و غیره) و برنامه‌های کاربردی مختلفی را با طراحی جدید وارد شبکه اصلی کرده است.

جداسازی صفحه داده از صفحه کنترل برخی از تهدیدات جزئی موجود در شبکه‌های سنتی را از بین می‌برد، اما نقطه‌ضعف‌های جدیدی (loophole) را برای SDN ایجاد کرده است.

از آنجایی‌که SDN برای برقراری ارتباط با زیرساخت به نرم‌افزارهای مختلف در لایه کاربرد متکی است، آسیب‌پذیری‌های موجود در کدها می‌تواند تأثیرات امنیتی جدی داشته باشد. قرار دادن کنترلر به عنوان یک عنصر مرکزی در بخش مدیریت و کنترل نیز منجر به ایجاد نقطه شکست برای کل زیرساخت می‌شود. کنترل‌کننده‌ها می‌توانند از چند سو (به‌عنوان‌مثال، رابط‌ها) تهدید شوند. واسط‌های جنوبی و شمالی به ترتیب درهای ورودی کنترل هستند که ممکن است هدف جذابی برای مهاجمان باشد. صفحه داده نیز هدف بالقوه مدل اتصال برای مهاجمان است. به دلیل تبادلات بین صفحه کنترل و صفحه داده، مهاجم می‌تواند جریان داده را با جریان سیل‌آسای[2] سوئیچ‌ها یا با دست‌کاری ارتباطات مختل کند. پروتکل OpenFlow که مسئول برقراری ارتباط بین دو صفحه است نیز می‌تواند در برابر برخی حملات آسیب‌پذیر باشد.

آسیب‌پذیری‌های SDN را می‌توان بر پنج محور اصلی تقسیم کرد[۶]:

۹-۱- لایه کاربرد

این‌ها آسیب‌پذیری‌های مرتبط با نرم‌افزار هستند که ممکن است از طریق حمله تزریق کد صورت گیرند. برخی از مطالعات روش‌های مقابله با مشکلات رایج لایه کاربرد را بررسی کرده‌اند. مشکلات موجود در لایه کاربرد عبارت‌اند از: چالش شناسایی و تطبیق قوانین جریان مغایر از برنامه‌های کاربردی OF (FortNOX) و مقابله با خرابی‌های برنامه‌های شبکه که منجر به از دست دادن کنترل شبکه می‌شود، و توانمندسازی انعطاف‌پذیری شبکه در برابر خرابی‌ها و خطاهای برنامه کاربردی SDN.

۹-۲- لایه کنترل

که شامل آسیب‌پذیری‌های مرتبط با کنترلر است. مطالعات بسیاری برای شناسایی و کاهش آسیب‌پذیری‌های کنترل‌کننده پیشنهاد شده است. عثمان و اوکامورا [۷] یک الگوریتم امضا[3] برای انتقال امن درخواست‌های نصب جریان از کنترل‌کننده به تجهیزات شبکه ارائه می‌کنند. در این سیستم از ارتباط امن با TLS استفاده می‌شود تا از دست‌کاری کانال کنترل جلوگیری شود. در [۸]، نویسندگان الگوریتمی را پیشنهاد کردند که یک معماری SDN امن و انعطاف‌پذیر را از طریق اجرای چندین کنترل‌کننده مقاوم در برابر خطا ارائه می‌دهد. در[۹] یک سیستم سلسله مراتبی متشکل از کنترلرها معرفی شده است که تأثیر برنامه مخربی که می‌تواند منجر به خرابی در صفحه کنترل شود را کاهش می‌دهد. مطالعات دیگری نیز برای تائید و اثبات امنیت و صحت صفحه کنترل SDN و حل مسائل مربوط به سیاست‌های امنیتی پیشنهاد شده است.

۹-۳- API ها

صفحه کنترل دو نوع API را نمایش می‌دهد: عمودی (یعنی جنوبی و شمالی) و افقی (شرقی ، غربی و رابط مدیریتی). موضوع نمایش رابط‌ها در [۱۰] مورد بحث قرار گرفته است. نویسندگان روش PermOF را پیشنهاد کردند که در آن حداقل سطح دسترسی را برای برنامه‌ها اختصاص می‌دهند تا بتوانند مجوزها را در ورودی API اعمال کنند. آثار مشابهی در [۱۱] معرفی شده است، که نویسندگان در آن به نبود امنیت در فرآیندهای ارتباطی بین برنامه‌های OpenFlow و صفحه کنترل اشاره کرده‌اند. به‌عنوان راه‌حل نیز کنترلر امنیتی تقویت

[1] Open Systems Interconnection model
[2] flooding
[3] signature algorithm

شده Floodlight پیشنهاد شده است که یک API شمالی احراز شده ارائه می‌دهد. سیستم دیگری نیز به نامFrenetic [۱۲] بر روی API شمالی جهت حل مغایرت‌های سیاستی تمرکز دارد.

۹-۴- پروتکل‌ها

در معماری SDN مبتنی بر OpenFlow، پروتکل OpenFlow را می‌توان طبق ارزیابی‌های آسیب‌پذیری در معرض تهدیدات مختلفی قرار داد، که ازجمله می‌توان به لایه امنیتی TLS اشاره کرد که در مشخصات پروتکل OpenFlow کاملاً اختیاری می‌باشد.

۹-۵- لایه زیرساخت

آسیب‌پذیری‌های مربوط به اجزای صفحه داده (به‌عنوان‌مثال سوئیچ‌ها و روترها) می‌توانند بردارهای حمله مختلفی مانند درج قوانین متقلبانه، اصلاح قوانین و جریان سیل‌آسا (flooding) ایجاد کنند. FlowChecker برای حل مسائلی معرفی شده است که مربوط به پیکربندی نادرست در سوئیچ‌های OpenFlow می‌باشد که باعث ایجاد واریانس در قوانین جریان می‌شود. Anteater نیز یک رویکرد آنالیز استاتیک برای تشخیص مشکلات پیکربندی شبکه معرفی می‌کند. روش OFHIP نیز برای فراهم آوردن تحرک و پویایی امن با OpenFlow جهت جلوگیری از اختلال در پردازش جریان، تخریب نشست (session) فعال TLS و مشکلات احراز هویت متقابل پیشنهاد شده است.

۱۰- طبقه‌بندی ترافیک

طبقه‌بندی ترافیک در بهینه‌سازی دسترسی به اینترنت و تجربه کاربری[1] بسیار مهم است. ازآنجایی‌که پهنای باند موجود محدود است، می‌توانیم با طبقه‌بندی ترافیک بهترین استفاده را از پهنای باند کنیم و ارائه دهندگان خدمات اینترنت نیز می‌توانند منابع را با اولویت‌بندی جریان بسته‌ها مدیریت کنند و جلوی حملات مخرب و بسته‌های مشکوک را بگیرند. در زیر انواع روش‌های طبقه‌بندی ترافیک شبکه توضیح داده شده است.[۱۳]

۱۰-۱- طبقه‌بندی ترافیک مبتنی بر پورت

طبقه‌بندی ترافیک را می‌توان با شناسایی کاربری‌های شبکه یا گروهی از آن‌ها به دست آورد. یکی از رویکردهای اصلی روش مبتنی بر پورت است. طبقه‌بندی مبتنی بر پورت یکی از ساده‌ترین تکنیک‌ها است، زیرا صرفاً هدر بسته را برای شناسایی شماره پورت و تطبیق آن با شماره پورت‌های شناخته‌شده بررسی می‌کند. برنامه‌های کاربردی شبکه، پورت‌های خود را از طریق مرجع واگذاری اعداد در *اینترنت* (IANA[2])به ثبت می‌رسانند. بااین‌حال، این رویکرد دارای محدودیت‌هایی نیز می‌باشد به‌عنوان‌مثال، برنامه‌ها می‌توانند از شماره پورت پویا یا پورت‌های سایر پروتکل‌ها برای پنهان شدن از ابزارهای امنیتی شبکه استفاده کنند. البته این روش به دلیل نتایج ضعیف طبقه‌بندی دیگر عملی نمی‌باشد، چراکه اکنون از پورت‌های پویا استفاده می‌شود.

۱۰-۲- طبقه‌بندی ترافیک مبتنی بر یادگیری ماشین

ظهور برنامه‌های کاربردی جدید و همچنین توسعه دستگاه‌های هوشمند، پیچیدگی و تنوع برنامه‌های آنلاین را افزایش داده است، همین امر مسئله طبقه‌بندی ترافیک را به یک کار دشوار تبدیل کرده است. جهت رفع محدودیت‌های DPI و طبقه‌بندی ترافیک مبتنی بر پورت، مطالعات بسیاری از روش‌های ML در طبقه‌بندی کاربرد شبکه استفاده کرده‌اند. این روش با بهره‌برداری از ویژگی‌های متمایز برنامه‌ها هنگام برقراری ارتباط در یک شبکه، به‌عنوان یک رویکرد جایگزین برای طبقه‌بندی ترافیک به حساب می‌آید. روش‌های یادگیری ماشین

[1] user experience
[2] Internet Assigned Network Authority

سعی می‌کنند الگوهای درونی درخواست‌ها را بر اساس مجموعه ویژگی‌های انتخاب شده شناسایی کنند. آن‌ها می‌توانند ترافیک رمزگذاری شده را طبقه‌بندی کرده و با هزینه محاسباتی کمتری به نتیجه برسند.

۱۱- تحقیقات انجام شده

روش‌ها و شیوه‌های مختلفی برای مقابله با حملات و نفوذکنندگان به سیستم‌ها و شبکه‌های کامپیوتری تدوین و ارائه شده است که معروف به روش‌های تشخیص نفوذ می‌باشند.

در شبکه‌های نرم‌افزار محور، بیشتر روش‌های تشخیص حمله DDoS موجود، به‌نوعی تبدیل روش‌های مورداستفاده در شبکه‌های سنتی است. الگوریتم مبتنی بر آنتروپی اطلاعات آماری یک روش رایج تشخیص DDoS است. این روش می‌تواند به‌سرعت حجم زیادی از داده‌های ترافیکی را با هزینه کم محاسبه کند، اما دقت آن به انتخاب آستانه بستگی دارد و یک‌طرفه بودن آن مشخص است.

کالکان و همکاران [۱۴] یک سیستم امتیازدهی مشترک مبتنی بر آنتروپی (JESS[1]) را برای شناسایی و کاهش حملات DDoS پیشنهاد کردند. از آنتروپی مشترک به‌عنوان ابزاری برای شناسایی حملات DDoS بدون افزایش قابل‌توجه بار کاری روی سوئیچ‌ها استفاده می‌کنند.

لیما و همکاران [۱۵] روشی را برای محافظت مؤثرتر از شبکه در برابر حملات DDoSاز طریق تجزیه و تحلیل آماری آنتروپی ترافیک معرفی کردند و مدلی را در Mininet برای تائید ارائه کردند. کومار و همکاران [۱۶] راه‌حلی ارائه کردند که می‌تواند به‌طور موثر حملات سیل SYN را در SDN شناسایی و کاهش دهد. که با محاسبه آنتروپی آدرس‌های IP مقصد شروع می‌شود، سپس از مجموعه‌ای از پرچم‌های TCP انتخاب شده به‌عنوان متغیرهای تصادفی استفاده می‌کند و درنهایت مهاجم را از طریق آستانه تطبیقی شناسایی می‌کند.

براگا و همکاران [۱۷] یک مقاله الگوریتم تشخیص حمله DDoS سبک وزن را بر اساس ویژگی‌های ترافیک پیشنهاد کرده‌اند. این الگوریتم از یک کنترل‌کننده NOX برای پردازش اطلاعات سوئیچ استفاده می‌کند و تجزیه و تحلیل ترافیک را بر اساس نقشه خودسازمان‌دهی (SOM) انجام می‌دهد. SOM یک شبکه عصبی مصنوعی یادگیری رقابتی و بدون نظارت است که می‌تواند به تشخیص DDoS سبک وزن دست یابد. علاوه بر این، الگوریتم k نزدیک‌ترین همسایه (KNN) نیز یک الگوریتم یادگیری ماشینی ساده و مؤثر است که جریان‌ها را با اندازه‌گیری فاصله انتزاعی بین بردارهای ویژگی ترافیک طبقه‌بندی می‌کند.

پنگ و همکاران [۱۸] الگوریتم تشخیص ترافیک غیرعادی، DPTCM-KNN را پیشنهاد کردند. این الگوریتم می‌تواند به‌طور مؤثری دقت تشخیص جریان غیرعادی را بهبود بخشد و درعین‌حال نرخ هشدار نادرست را در فرآیند تشخیص DDoS کاهش دهد. اگرچه بسیاری از محققان راه‌حل‌های مختلفی را بر اساس الگوریتم‌های یادگیری ماشین برای تشخیص DDoS در SDN ارائه کرده‌اند، اما این روش‌ها همچنان در دقت و کارایی مشکل دارند.

زنگ و همکاران [۱۹] از ویژگی‌های جریان شروع کردند و مجموعه‌ای از شاخص‌های جریان دقیق‌تر و جامع‌تر را پیشنهاد کردند. نویسندگان ۹ ویژگی منفرد و ۳۹ ویژگی دوگانه را از ابعاد مختلف مانند زمان، مکان، دسته و شدت استخراج کردند تا طیف ویژگی‌های رفتار ترافیک آدرس IP را تشکیل دهند. ویژگی‌های ترافیکی ریز به میزان زیادی دقت تشخیص را بهبود می‌بخشد.

خو و همکاران [۲۰] الگوریتم تشخیص DDoS را بهبود بخشیدند. آن‌ها یک روش تشخیص DDoS بر اساس K-FKNN و یک سیستم تشخیص ماژول را برای بهبود کارایی و دقت تشخیص پیشنهاد کردند. علاوه بر این، برخی از محققان با بهبود روش جمع‌آوری داده‌های جریان، سربار کانال بین صفحه داده و صفحه کنترل را کاهش دادند و کارایی تشخیص را بهبود بخشیدند. در پژوهش انجام‌شده در مقاله [۲۱]، نویسنده از فناوری sFlow برای نمونه‌برداری از ترافیک شبکه استفاده می‌کند، که با روش جمع‌آوری جدول جریان خود SDN بر اساس پروتکل Open Flow مقایسه می‌شود و به‌طور موثر سربار صفحه کنترل را کاهش می‌دهد.

[1] joint entropy-based security scheme

شبکه‌های عصبی مصنوعی در تحقیقات شناسایی حملات DDoS شناخته‌شده و ناشناخته استفاده شده است که نشان می‌دهد ما می‌توانیم حمله DDoS به کنترل‌کننده SDN را با دقت قابل‌توجهی شناسایی کنیم و از آسیب جدی به کنترل‌کننده جلوگیری کنیم. شبکه عصبی پرسپترون در مقاله [۲۲] استفاده شد، و نتایج ارزیابی نشان داد که بهبود قابل‌توجهی در میزان تشخیص به دست آمد درحالی‌که کاهش نرخ هشدار کاذب نیز در مقایسه با نزدیک‌ترین کار قبلی به دست آمد. علاوه بر این، سیستم آن‌ها توانست میانگین زمان تشخیص را در سطح قابل قبولی حفظ کند. آن‌ها یک روش کارآمد را برای کاهش حمله برای کار آینده بررسی می‌کنند.

در پژوهش انجام شده توسط Kokila و همکاران [۲۳] ماشین بردار پشتیبانی (SVM) به دلیل دقت بالا و نرخ مثبت کاذب کمتر برای طبقه‌بندی حمله DDoS با ترافیک معمولی استفاده شده است. طبقه‌بندی‌کننده SVM با سایر طبقه‌بندی‌کننده‌ها برای تشخیص حمله DDoS مقایسه شد و SVM طبقه‌بندی دقیقی نسبت به سایر تکنیک‌ها ارائه کرد. شناسایی بلادرنگ DDoS و ادغام الگوی ترافیک ساخته شده در SVM با کنترلر SDN کار آینده آن‌ها بود.

در مقاله ارائه شده توسط ون و همکاران [۲۴] از منطق فازی برای تشخیص ترافیک واقعی حمله DDoS در SDN استفاده شده است. نویسندگان مشکلات موجود پروتکل OpenFlow را حل کرده‌اند. آن‌ها الگوریتم کاهش DDoS مبتنی بر منطق فازی را پیشنهاد کردند که معیارهای متعددی را برای تشخیص DDoS به کار می‌برد. سیستم آن‌ها توانایی شناسایی و فیلتر کردن ۹۷٪ از جریان‌های حمله با نرخ مثبت کاذب ۵٪ را نشان داد. آن‌ها مایلند پروتکل OpenFlow را برای دستیابی به عملکرد قوی و سریع‌تر گسترش دهند.

در پژوهش انجام شده توسط وو و همکاران[۲۵]، محققان سیستمی را برای شناسایی حملات DDoS بر اساس تکنیک درخت تصمیم طراحی کرده‌اند و آن‌ها را با تکنیک تطبیق الگوی جریان ترافیک به مکان‌های تقریبی مهاجم ردیابی کردند . سیستم آن‌ها می‌تواند حمله را با نسبت مثبت کاذب ۱٫۲٪ تا ۲٫۴٪ تشخیص دهد. آن‌ها آزمایش خود را بر روی سیستم DETER انجام دادند. نتایج آن‌ها نشان داد که سیستم پیشنهادی آن‌ها قادر به شناسایی حملات و ردیابی با دقت بالایی است. الگوریتم‌های تکاملی (EAs) برای تشخیص حمله DDoS ارائه شده است. محققان چهار نوع EA را که به‌طور گسترده در SDN های کنونی کاربرد دارند، بررسی کردند: الگوریتم‌های ژنتیک (GAs)، بهینه‌سازی ازدحام ذرات (PSO)، بهینه‌سازی کلونی مورچه‌ها (ACO)، و آنیل شبیه‌سازی شده (SA). هر چهار EA با هم مقایسه شدند و کاربردهای این چهار EA در SDN دسته‌بندی شدند.

در پژوهش انجام شده توسط سینگ و همکاران [۲۶] به‌منظور دستیابی به یک تکنیک تشخیص DDoS خوب، محققان با استفاده از تجزیه و تحلیل ویژگی‌ها راه‌حل بهتری برای تشخیص‌ها ارائه کرده‌اند. در روش ارائه شده از الگوریتم طبقه‌بندی کننده بیز ساده برای طبقه‌بندی بسته‌ها به بسته‌های عادی و بسته‌های حمله استفاده شد. استفاده از الگوریتم کسب اطلاعات باعث افزایش کارایی می‌شود. مجموعه داده‌های CAIDA 2008 و CAIDA Trace 2015 ناشناس برای انتخاب ویژگی و طبقه‌بندی آن‌ها استفاده شد.

در مقاله انجام شده توسط پرامانا و همکاران [۲۷] روشی برای شناسایی یک حمله DoS با استفاده از تکنیک خوشه‌بندی با الگوریتم k-means که برای اصلاح و توسعه به طرق ممکن در دسترس است ارائه شده است. با استفاده از این الگوریتم، نتایج آن‌ها بر روی نرخ تشخیص، دقت و نرخ مثبت کاذب ارزیابی شد. روش آن‌ها با استفاده از مجموعه داده DARPA 98 با نتیجه رضایت‌بخش ارزیابی شده است. در آینده، آن‌ها مایلند در به حداقل رساندن نرخ مثبت کاذب پیشرفت کنند.

۱۲- نتیجه‌گیری

با محبوبیت اینترنت و برنامه‌های کاربردی مرتبط، تعداد کاربران شبکه به طور چشمگیری افزایش یافته است. طی سال‌های اخیر، شاهد افزایش قابل توجهی در تعداد و پیچیدگی حملات سایبری بوده‌ایم که کاربران خانگی، مشاغل، سازمان‌های دولتی و حتی زیرساخت‌های حیاتی را هدف قرار می‌دهند. در بسیاری از موارد، شناسایی حملات در مراحل اولیه بسیار مهم است، قبل از اینکه آسیب قابل توجهی به شبکه‌ها و سیستم‌های محافظت شده از جمله دسترسی به داده‌های حساس وارد شود. برای این منظور، محققان و متخصصان امنیت

سایبری در حال بررسی استفاده از فناوری شبکه‌های نرم‌افزار محور برای دفاع کارآمد و در زمان واقعی در برابر حملات سایبری هستند. شبکه نرم‌افزار محور به روند واقعی مدل خدمات فناوری اطلاعات تبدیل شده است که یک راه‌حل پردازش مقرون به صرفه و مقیاس‌پذیر ارائه می‌دهد. شبکه‌های نرم‌افزار محور با جدا کردن صفحه کنترل از صفحه داده، به طور منطقی متمرکز شدند. این ویژگی قابلیت برنامه‌ریزی شبکه را فعال می‌کند و این پتانسیل را دارد که تقریباً بلافاصله ترافیک شبکه را در صورت شناسایی برخی فعالیت‌های مخرب مسدود کند.

معماری شبکه نرم‌افزار محور شامل یک کنترلر شبکه متمرکز با نمای کلی از شبکه و یک رابط برنامه‌نویسی کاربردی برای توسعه برنامه‌های کاربردی شبکه است. از جمله مزایای شبکه نرم‌افزار محور امکان حذف تقریباً فوری ترافیک مخرب از رابط شبکه پس از شناسایی است.

هدف از طراحی سیستم تشخیص ناهنجاری، شناسایی طرح‌های ترافیکی غیرعادی(حمله) در شبکه و جلوگیری از نفوذ و خرابکاری در شبکه است. در زمانی که سازمان یا افرادی قصد انجام حمله به یک شبکه را داشته باشند با توجه به نوع حمله تغییراتی در ترافیک شبکه ایجاد می‌شود و اگر این تغییرات از قبل کشف و شناسایی شود می‌توان وجود ناهنجاری و حمله را با برچسب‌گذاری جریان ترافیکی به‌صورت عادی یا ناهنجاری و استفاده از الگوریتم‌های طبقه‌بندی تشخیص داد. چشم‌انداز جهانی شبکه می‌تواند امنیت سیستم‌ها را بهبود بخشد. این امنیت را نمی‌توان تنها در امنیت میزبان مستقر کرد، زیرا چنین دفاعی زمانی که میزبان در معرض خطر قرار می‌گیرد بی‌اثر است. سیستم مبتنی بر OpenFlow به کنترل‌کننده امکان تجزیه‌وتحلیل و تائید اتصالات و جریان ترافیک در شبکه را می‌دهد. روشی که در این پایان‌نامه بررسی و تحلیل خواهد شد، استفاده از طبقه‌بندی ترافیک‌ها با کمک یادگیری عمیق می‌باشد. می‌توان از طبقه‌بندی ترافیک‌ها برای تشخیص ناهنجاری‌های امنیتی در شبکه‌ها استفاده کرد.

بحث امنیت شبکه‌های نرم‌افزار محور به دلیل متمرکز بودن کنترل کل شبکه بسیار اهمیت دارد. از این‌رو محققان توجه بسیار زیادی به امر افزایش امنیت این شبکه‌ها دارند. ما در این پژوهش به کارهای صورت گرفته در زمینه تشخیص ناهنجاری در شبکه‌های نرم‌افزار محور پرداخته‌ایم. در تحقیقات صورت گرفته، حمله‌های انکار از سرویس مهم‌ترین تهدید خارجی در شبکه‌های SDN عنوان شده است. جمع‌آوری داده با استفاده از پروتکل بومی OpenFlow در شبکه‌های با ترافیک بالا باعث اشباع سطح کنترل می‌شود، بنابراین استفاده از پروتکل‌هایی که وظیفه جمع‌آوری داده را به عهده می‌گیرد برای شبکه‌های با ترافیک بالا ضروری است. همچنین در مقالات ارائه شده از روش‌های مبتنی بر پیش‌بینی(یادگیری ماشین و غیره) نیز برای تشخیص ناهنجاری استفاده شده است، مزیت این روش‌ها نسبت به دیگر روش‌های دیگر جامع‌تر بودن آن‌ها و دقت بالاتر آن‌ها است.

مراجع

[1] A. Mestres et al., "Knowledge-defined networking" *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, pp. 2-10, 2017.

[2] S. Chibani and F.-X. Coudert, "Machine learning approaches for the prediction of materials properties," *APL Materials*, vol. 8, no. 8, p. 080701, 2020.

[3] R. M. AlZoman and M. J. Alenazi, "A comparative study of traffic classification techniques for smart city networks," Sensors, vol. 21, no. 14, p. 4677, 2021.

[4] X. Luo, X. Li, Z. Wang, and J. Liang, "Discriminant autoencoder for feature extraction in fault diagnosis," *Chemometrics and Intelligent Laboratory Systems*, vol. 192, pp. 103814, 2019.

[5] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51-62, 2020.

[6] K. Benzekki, A. El Fergougui, and A. Elbelrhiti Elalaoui, "Software-defined networking (SDN): a survey," *Security and communication networks*, vol. 9, no. 18, pp. 5803-5833, 2016.

[7] O. M. Othman and K. Okamura, "Securing distributed control of software defined networks," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 13, no. 9, p. 5, 2013.

[8] H. Li, P. Li, S. Guo, and A. Nayak, "Byzantine-resilient secure software-defined networks with multiple controllers in cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 4, pp. 436-447, 2014.

[9] D. Yu, *Authentication for resilience: the case of SDN (transcript of discussion)*, in Cambridge international workshop on security protocols, Springer, pp. 45-53, 2013.

[10] X. Wen, Y. Chen, C. Hu, C. Shi, and Y. Wang, "Towards a secure controller platform for openflow applications," *in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pp. 171-172, 2013.
[11] P. A. Porras, S. Cheung, M. W. Fong, K. Skinner, and V. Yegneswaran, "Securing the software defined network control layer," *in NDSS*, 2015.
[12] N. Foster et al., "Frenetic: A network programming language," *ACM Sigplan Notices*, vol. 46, no. 9, pp. 279-291, 2011.
[13] O. Aouedi, K. Piamrat, and D. Bagadthey, "A semi-supervised stacked autoencoder approach for network traffic classification," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, IEEE, pp. 1-6, 2020.
[14] K. Kalkan, L. Altay, G. Gür, and F. Alagöz, "JESS: Joint entropy-based DDoS defense scheme in SDN," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2358-2372, 2018.
[15] N. A. Lima and M. P. Fernandez, "Towards an efficient DDoS detection scheme for software-defined networks," IEEE Latin America Transactions, vol. 16, no. 8, pp. 2296-2301, 2018.
[16] P. Kumar, M. Tripathi, A. Nehra, M. Conti, and C. Lal, "SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1545-1559, 2018.
[17] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," *in IEEE Local Computer Network Conference*, IEEE, pp. 408-415, 2010.
[18] H. Peng, Z. Sun, X. Zhao, S. Tan, and Z. Sun, "A detection method for anomaly flow in software defined network," *IEEE Access*, vol. 6, pp. 27809-27817, 2018.
[19] X.-D. Zang, J. Gong, and X.-Y. Hu, "An adaptive profile-based approach for detecting anomalous traffic in backbone," *IEEE Access*, vol. 7, pp. 56920-56934, 2019.
[20] Y. Xu, H. Sun, F. Xiang, and Z. Sun, "Efficient DDoS detection based on K-FKNN in software defined networks," *IEEE access*, vol. 7, pp. 160536-160545, 2019.
[21] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Computer Networks*, vol. 62, pp. 122-136, 2014.
[22] F. Gharvirian and A. Bohlooli, "Neural network based protection of software defined network controller against distributed denial of service attacks," *International Journal of Engineering*, vol. 30, no. 11, pp. 1714-1722, 2017.
[23] R. Kokila, S. T. Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," *in 2014 sixth international conference on advanced computing (ICoAC)*, IEEE, pp. 205-210, 2014.
[24] T. Dang-Van and H. Truong-Thu, "A multi-criteria based software defined networking system Architecture for DDoS-attack mitigation," *REV Journal on Electronics and Communications*, vol. 6, no. 3-4, 2017.
[25] Y.-C. Wu, H.-R. Tseng, W. Yang, and R.-H. Jan, "DDoS detection and traceback with decision tree and grey relational analysis," *in 2009 Third International Conference on Multimedia and Ubiquitous Engineering*, IEEE, pp. 306-314, 2009.
[26] N. A. Singh, K. J. Singh, and T. De, "Distributed denial of service attack detection using naive Bayes classifier through info gain feature selection," *in Proceedings of the International Conference on Informatics and Analytics*, pp. 1-9, 2016.
[27] M. I. W. Pramana, Y. Purwanto, and F. Y. Suratman, "DDoS detection using modified K-means clustering with chain initialization over landmark window," *in 2015 International Conference on Control, Electronics*, Renewable Energy and Communications (ICCEREC), IEEE, pp. 7-11, 2015.