ITRC
ICT Research Institute
(Iran Telecommunication Research Center)

Journal of Information and Communication Technology

Iranian Society of Information and Communication Technology

Volume 17, Issue 63-64, Spring and Summer 2025, pp. 110-122

Detecting Distributed Denial of Service Attacks in Software-Defined Networks with a Deep Learning Approach

Younes Mehdizadeh[1*], Mehdi sadegh zadeh[2]

[1] Department of Information Technology Management, Faculty of Management and Economics, Islamic Azad University, Science and Research Branch, Tehran, Iran

[2] Department of Computer Science, Islamic Azad University, Science and Research Branch, Tehran, Iran

Received: 27 January 2024, Revised: 6 January 2025, Accepted: 29 January 2025
Paper type: Research

Abstract

The growth of cloud computing has led to the development of software-defined networks. These networks enable dynamic management and performance improvement. Security threats in this type of network are a growing concern. Especially, the controller of these networks is an attractive target for hackers and distributed denial of service attacks. Many researchers have proposed different methods to detect these attacks, whose false detection rate is very high and has led to a decrease in detection accuracy. For this purpose, in this research, the focus is on detecting distributed denial of service attacks through deep learning using prominent features of packets. After pre-processing and preparing the data, the proposed method separates the salient and important features of the packages through the support vector machine method and finally by using an innovative hybrid neural network consisting of convolutional neural network, sample recurrent neural network and Long-term short-term memory neural network separates attack packets from normal packets. A standard data set has been used to evaluate the proposed method through standard evaluation criteria such as detection accuracy, precision, false detection rate and harmonic mean accuracy. The findings show that the proposed method detects distributed denial of service attacks with 95.2% detection accuracy, 92.09% precision, 5.1% false alarm rate, and 93.87% F1_measure.

Keywords: Software-Defined Networks, Distributed Denial of Service Attacks, Cloud Computing, Deep Learning, Neural Networks.

[*] Corresponding Author's email: ymz.info1989@gmail.com

دوفصلنامه
فناوری اطلاعات و ارتباطات ایران

سال هفدهم، شماره‌های ۶۳ و ۶۴ ، بهار و تابستان ۱۴۰۴، صفحه ۱۱۰ الی ۱۲۲

تشخیص حملات انکار سرویس توزیع شده در شبکه‌های مبتنی بر نرم‌افزار با رویکرد یادگیری عمیق

یونس مهدی‌زاده[1]*، مهدی صادق‌زاده[2]

[1] گروه مدیریت فناوری اطلاعات، دانشکده مدیریت و اقتصاد، دانشگاه آزاد اسلامی واحد علوم و تحقیقات، تهران،ایران

[2] دانشیار، گروه کامپیوتر، دانشکده کامپیوتر، دانشگاه آزاد اسلامی واحد علوم و تحقیقات، تهران، ایران

تاریخ دریافت: ۱۴۰۲/۱۱/۰۷ تاریخ بازبینی: ۱۴۰۳/۱۰/۱۷ تاریخ پذیرش: ۱۴۰۳/۱۱/۱۰

نوع مقاله: پژوهشی

چکیده

رشد محاسبات ابری منجر به توسعه شبکه‌های مبتنی بر نرم‌افزار شده است. تهدیدات امنیتی در این نوع شبکه یک نگرانی است. کنترل‌کننده این شبکه‌ها، هدف جذابی برای حملات انکار سرویس توزیع شده است. پژوهشگران روش‌های مختلفی را برای شناسایی این حملات ارائه کرده‌اند که آمار تشخیص اشتباه آنها بسیار بالا است. به همین منظور این پژوهش، روشی برای تشخیص حملات انکار سرویس توزیع شده از طریق یادگیری عمیق با استفاده از ویژگی‌های برجسته بسته‌ها، پیشنهاد می‌کند. روش پیشنهادی پس از پیش پردازش و آماده‌سازی داده‌ها، ویژگی‌های با اهمیت بسته‌ها را از طریق روش ماشین بردار پشتیبان، جدا می‌کند و نهایتا با یک شبکه عصبی ترکیبی ابتکاری متشکل از یک شبکه عصبی کانولوشنال، دو نوع شبکه بازگشتی ساده و حافظه کوتاه و بلند مدت، بسته‌های حمله را از بسته‌های عادی جدا می‌کند. ارزیابی عملکرد روش پیشنهادی از طریق معیارهای استاندارد روی یک مجموعه داده استاندارد انجام می‌شود. یافته‌ها نشان می‌دهد روش پیشنهادی حملات انکار سرویس توزیع شده را با دقت تشخیص ۹۵,۲ درصد، حساسیت ۹۲,۰۹درصد، نرخ تشخیص اشتباه ۲,۷درصد ومیانگین هارمونیک دقت ۹۳,۸۷درصد تشخیص می‌دهد.

کلیدواژگان: شبکه‌های مبتنی بر نرم‌افزار، حملات انکار سرویس توزیع شده، محاسبات ابری، یادگیری عمیق، شبکه‌های عصبی.

* رایانامهٔ نویسنده مسؤول: ymz.info1989@gmail.com

۱- مقدمه

شبکه مبتنی بر نرم‌افزار[1] یک رویکرد انقلابی برای مدیریت شبکه است که به مدیران شبکه اجازه می‌دهد تا کل زیرساخت شبکه را از طریق برنامه‌های کاربردی نرم‌افزاری و سیاست‌های قابل برنامه‌ریزی، کنترل و مدیریت کنند. این تغییر پارادایم در شبکه، انعطاف‌پذیری، مقیاس‌پذیری و کارایی بیشتری را در مقایسه با مدل‌های شبکه سنتی سخت‌افزارمحور ارائه می‌دهد [۱].

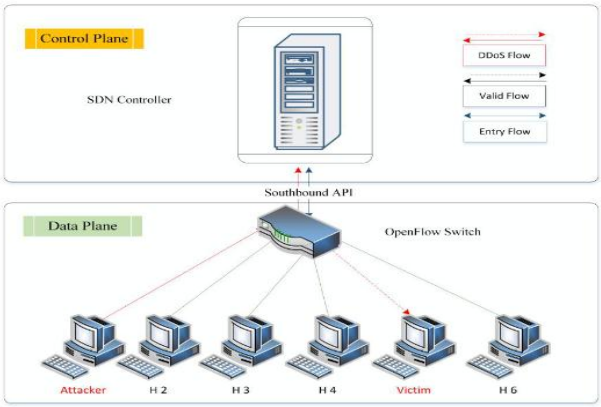
شبکه مبتنی بر نرم‌افزار به مدیران شبکه اجازه می‌دهد تا رفتار شبکه را از طریق نرم‌افزار، برنامه‌ریزی کنند. این قابلیت برنامه‌ریزی، تغییرات سریع پیکربندی، سازگاری با شرایط مختلف و توانایی اجرای سیاست‌ها به صورت پویا را امکان‌پذیر می‌کند.

شبکه مبتنی بر نرم‌افزار، استانداردهای باز را ترویج و رابط‌های برنامه‌نویسی کاربردی[2] را فراهم می‌کند که به برنامه‌های شخص ثالث اجازه می‌دهد با زیرساخت شبکه تعامل داشته باشند. قابلیت بازبودن، باعث تقویت نوآوری، تشویق توسعه برنامه‌های کاربردی جدید و تسهیل همکاری بین فروشندگان تجهیزات مختلف می‌شود [۱].

شبکه مبتنی بر نرم‌افزار، امکان ایجاد چندین شبکه مجازی را روی یک زیرساخت فیزیکی فراهم می‌کند. این قابلیت برای محیط‌های ابری، مراکز داده و سناریوهایی که استفاده بهینه از منابع ضروری است، بسیار مفید می‌باشد.

در معماری‌های شبکه سنتی، سطح کنترل که تصمیم‌گیری در مورد نحوه ارسال داده‌ها و سطح داده که به ارسال بسته‌های داده اشاره دارد، در دستگاه‌های شبکه مانند سوئیچ‌ها و روترها ادغام می‌شوند. شبکه مبتنی بر نرم‌افزار این دو سطح را جدا می‌کند و کنترل را در یک کنترل‌کننده[3] مبتنی بر نرم‌افزار متمرکز می‌کند. کنترل‌کننده متمرکز به عنوان مغز شبکه عمل می‌کند. این کنترل‌کننده با سوئیچ‌ها و مسیریاب‌ها در شبکه ارتباط برقرار می‌کند و دیدی جامع از شبکه ارائه می‌دهد تا بر اساس سیاست‌های تعریف شده در مورد نحوه مدیریت ترافیک تصمیم‌گیری شود.[۲]

حملات انکار سرویس توزیع شده[4] از ماشین‌های در معرض خطر متعدد برای هجوم ترافیک به کنترل‌کننده استفاده می‌کند و باعث افزایش بار روی آن می‌شود . با توجه به اینکه کنترل‌کننده قادر به تشخیص تمایز بین ترافیک عادی و غیر عادی شبکه نیست [۳] و [۴]، حملات انکار سرویس توزیع شده، به مرور زمان باعث از کار افتادن کنترل‌کننده می‌شوند. شکل ۱ نحوه حملات انکار سرویس توزیع شده را به به شبکه مبتنی بر نرم‌افزار نشان می‌دهد.



شکل ۱. حمله انکار سرویس توزیع شده به کنترل‌کننده شبکه مبتنی بر نرم‌افزار[۱۰]

پژوهشگران بسیاری پیشنهادات مختلفی را برای تشخیص حملات انکار سرویس به کنترل‌کننده شبکه مبتنی بر نرم‌افزار ارائه کرده‌اند [۵]تا[۹] با وجود کارایی این پیشنهادات، شناسایی حملات انکار سرویس توزیع شده به کنترل‌کننده‌های شبکه مبتنی بر نرم‌افزار همچنان یک مشکل چالش بر انگیز است[۱۰]. زیرا اکثر این پیشنهادات از مشکل مثبت کاذب[5] بالا رنج می‌برند. منظور از مثبت کاذب تعداد تشخیص اشتباه بسته‌های عادی شبکه به عنوان بسته غیر عادی یا حمله است. این دقت پایین را می‌توان به دو عامل اصلی نسبت داد: الف- اتکا به ویژگی‌های غیر مرتبط بسته‌ها[۶] ب- ارزیابی با استفاده از داده‌های غیر واقعی[۷].

منصور و همکاران[۱۰] استفاده از دو مکانیزم نرخ گین اطلاعات و انتخاب ویژگی مبتنی بر خی-دو را برای انتخاب ویژگی‌های برجسته برای تشخیص بسته‌های عادی و حمله استفاده می‌کنند. این مقاله استفاده از روش ماشین بردار پشتیبان[6] را برای انتخاب ویژگی‌های برجسته پیشنهاد می‌کند. این روش تاثیر بالایی بر افزایش قدرت تشخیص، در مقایسه با مکانیزم‌های قبلی از خود نشان می‌دهد.

استفاده از شبکه‌های عصبی عمیق مختلف می‌تواند دقت تشخیص[7] حملات را افزایش دهد. پژوهش‌های فراوانی با استفاده از انواع شبکه‌های عصبی مانند شبکه‌های عصبی بازگشتی ساده[8]،

[1] Software Defined network[SDN]
[2] Application Programming Interface [API]
[3] Controller
[4] Distribute denial of service (DDOS)
[5] False Positive [FP]
[6] Support Vector Machine (SVM)
[7] Detection Accuracy (DA)
[8] Sample Recurrent Neural Networks (S_RNNs)

شبکه‌های عصبی کانولوشنال[1] و شبکه‌های با حافظه کوتاه‌مدت و بلندمدت[2] به صورت جداگانه برای افزایش قدرت تشخیص حملات انجام شده است [۱۱].روش پیشنهادی از ترکیب هر سه نوع شبکه عصبی برای ایجاد مدل استفاده می‌کند. ارزیابی مدل نشان می‌دهد که روش پیشنهادی باعث بهبود معیارها می‌شود.

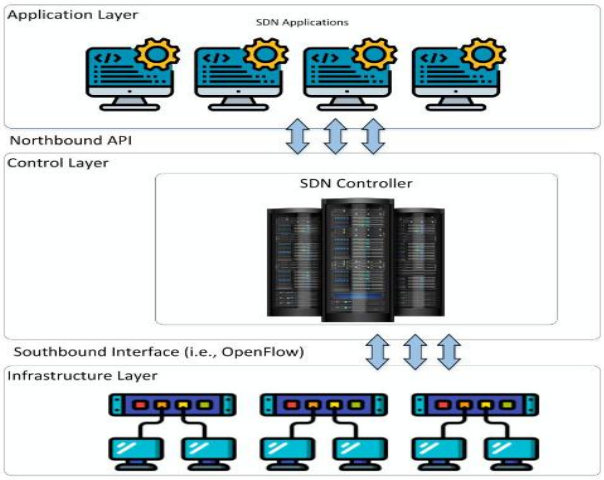
در ادامه مقاله ابتدا در بخش ۲ پیشینه تحقیق مورد بررسی قرار می‌گیرد. در بخش ۳ کارهای مرتبط مورد بررسی قرار می‌گیرد. بخش ۴ روش پیشنهادی معرفی می‌شود، در بخش ۵ نتایج آزمایش نشان داده می‌شود . بخش ۶ بحث و نتیجه‌گیری صورت می‌پذیرد و در بخش ۷ جهت‌گیری تحقیقات آینده ارئه می‌شود.

۲- پیشینه تحقیق

در این بخش نمای کلی معماری شبکه مبتنی بر نرم‌افزار ارائه می‌شود.نقش کنترل‌کننده‌ها و پروتکل جریان باز[3] مورد بحث قرار می‌گیرد. همچنین اهمیت رویکردهای مبتنی بر یادگیری عمیق در شبکه‌های مبتنی بر نرم‌افزار بیان می‌شود.

۲-۱- شبکه مبتنی بر نرم‌افزار

شبکه‌های سنتی به دلیل عدم انعطاف پذیری کافی مقرون به صرفه نیستند و برای پاسخگویی به نیازهای اینترنت فعلی مناسب نخواهند بود [۱۲]. بنابراین ظهور نسل جدیدی از شبکه که بتواند پاسخگوی نیازها و دارای قابلیت انعطاف پذیری باشد، مطرح می‌شود. شبکه مبتنی بر نرم‌افزار امکان دست یافتن به یک شبکه قابل برنامه ریزی با جدا کردن سطح کنترل از سطح داده، برای بهبود کارایی شبکه را فراهم می‌کند [۱۳]. این جداسازی مزایایی همچون مدیریت ساده شبکه، بهبود کارایی شبکه و ایجاد نوآوری در شبکه را فراهم می‌کند. سطح کنترل، اطلاعات لازم برای مسیریابی در شبکه را فراهم می‌کند. سطح داده نیز وظیفه انتقال بسته‌ها از درگاه ورودی به درگاه خروجی بر اساس اطلاعات درج شده در جدول مسیریابی خود را بر عهده دارد[۱۰]. در شبکه مبتنی بر نرم‌افزار سطح جداشده کنترل در سرور یا برنامه‌ای به نام کنترل‌کننده قرار می‌گیرد[۱۰]. سطح داده نیز در سوئیچ یا مسیریاب به عنوان ارسال‌کننده باقی می‌ماند[۱۰]. پیدایش شبکه مبتنی بر نرم‌افزار توجه تعداد زیادی از دانشگاه‌ها و صنایع را به پیاده‌سازی آن در زیرساخت‌های ارتباطی خود معطوف ساخته است[۱۴]. روش‌های پیکربندی این شبکه‌ها ساده‌تر و دقیق‌تر است و امکان بهره‌گیری بیشتر از زیرساخت‌های فیزیکی را فراهم کرده است [۱۵]. گروهی از اپراتورهای شبکه، ارائه دهندگان خدمات شبکه و تولیدکنندگان تجهیزات شبکه، سازمانی به نام بنیاد شبکه باز[4] [۱۶] ایجاد کردند تا به ترویج شبکه‌های مبتنی بر نرم‌افزار و ارائه یک پروتکل ارتباطی استاندارد برای این شبکه بپردازند[۱۹].



شکل ۲. معماری شبکه مبتنی بر نرم‌افزار[۱۰]

در شبکه‌های مبتنی بر نرم‌افزار، کنترل‌کننده مسئول انتشار هر جریان در شبکه است که این عمل را با تخصیص جریان‌های ورودی به سوئیچ‌ها انجام می‌دهد[۱۷]. این وظیفه، نقشی محوری به کنترل‌کننده بخشیده است؛ چراکه به واسطه آن می‌توان دانش کاملی از شبکه را در بهینه‌سازی مدیریت جریان و پشتیبانی از نیاز کاربر به خدمت گرفت[۱۸]. یکی از معایب این نوع شبکه وابستگی شدید در دسترس بودن[5] شبکه به عملکرد کنترل‌کننده است. از این رو حفاظت از کنترل‌کننده در برابر حملات امنیتی بسیار حائز اهمیت است. شکل ۲ معماری شبکه مبتنی بر نرم‌افزار را نشان می‌دهد.

۲-۲- پروتکل جریان باز

جریان باز یک پروتکل ارتباطی است که کنترل‌کننده شبکه مبتنی بر نرم‌افزار را قادر می‌سازد تا با عناصر ارسال و دریافت در یک شبکه مانند سوئیچ‌ها و روترها تعامل داشته باشد. این پروتکل امکان کنترل متمرکز روی شبکه را فراهم می‌کند و پیکربندی پویا و مدیریت دستگاه‌های شبکه را ممکن می‌سازد[۳]. عملیات بین کنترل‌کننده و سوئیچ‌ها به این صورت است که سوئیچ پس از راه‌اندازی، اتصالی

[1] Convolutional Neural Networks (CNNs)
[2] Long Short-Term Memory (LSTM) Networks
[3] Open Flow
[4] Open Network Foundation
[5] Availability

با کنترل‌کننده برقرار می‌کند که به آن کانال جریان باز می‌گویند. بسته به اینکه چه کسی اولین حرکت را انجام دهد حالت کنشی[1] و واکنشی[2] به وجود می‌آید [۱۹].

ورودی‌های جریان از پیش نصب شده کنترل‌کننده در سوئیچ‌ها حالت کنشی نامیده می‌شوند. این حالت می‌تواند در کل زمان کار رخ دهد و مستلزم ارسال ورودی جریان مستقیم پس از ایجاد کانال جریان باز نیست. حالت واکنشی بالعکس است[۱۹]. هنگامی که سوئیچ‌های جریان باز یک بسته جدید دریافت می‌کنند که با هیچ ورودی جریانی مطابقت ندارد، آن را در یک پیام بسته_ ورودی[3] کپسوله می‌کنند و آن را از طریق کانال جریال باز به کنترل‌کننده ارسال می‌کنند. کنترل‌کننده با یک جدول جریان[4] و یک بسته اصلی[5] پاسخ می‌دهد[۲۰].

۲-۳- رویکرد یادگیری عمیق[6]

هوش مصنوعی یکی از زمینه‌های علمی درحال توسعه است که کاربردهای عملی آن باعث ایجاد انگیزه برای تدوام پژوهش شده است[۱۰]. توسعه نرم‌افزارهای هوشمند باعث خودکارسازی وظایف، تجزیه و تحلیل تصاویر، درک مکالمات، انجام تشخیص‌های پزشکی، ایجاد زیرساخت‌های هوشمند، توانمندسازی افراد دارای معلولیت جسمی و حمایت از تحقیقات علمی شده است [۱۰]. یادگیری ماشین[7] پایه و اساس اکثر راه حل‌های هوش مصنوعی است. با حجم وسیعی از داده‌های موجود امروزه، می‌توانیم از آن برای آموزش مدل‌هایی استفاده کنیم که می‌توانند بر اساس الگوها و ارتباط‌های موجود در داده‌ها، پیش‌بینی و استنتاج کنند. هوش مصنوعی ،یادگیری ماشین و یادگیری عمیق رابطه‌ای مشترک دارند که در آن یادگیری عمیق نوعی یادگیری بازنمایی[8] است، و یادگیری ماشین در بسیاری از سیستم‌های هوش مصنوعی، استفاده می‌شود [۱۰].

یادگیری عمیق قابلیت‌های یادگیری ماشین کلاسیک را با افزایش پیچیدگی مدل و اصلاح عملیات داده برای فعال‌سازی نمایش سلسله مراتبی داده از طریق چندین لایه انتزاعی افزایش می‌دهد [۲۱، ۲۲]. یکی از مزایای کلیدی یادگیری عمیق، یادگیری ویژگی است، که در آن ویژگی‌های داده خام به طور خودکار استخراج می‌شوند و ویژگی‌های سطح بالاتر از طریق ترکیب ویژگی‌های سطح پایین شکل می‌گیرند [۲۳]. یادگیری عمیق در رسیدگی به مسائل و مدل‌های پیچیده برتری دارد و امکان موازی‌سازی کارآمد را فراهم می‌کند. اجزای یادگیری عمیق، بسته به معماری شبکه مورد استفاده، متفاوت است و ممکن است شامل لایه‌های ادغام، کانولوشن، گیت، لایه‌های کاملاً متصل، توابع فعال‌ساز، سلول‌های حافظه و روش‌های رمزگذاری/رمزگشایی باشد [۲۴].

جدول ۱. مقایسه سه نوع شبکه عصبی

معیارها	شبکه عصبی کانولوشنال	شبکه عصبی بازگشتی ساده	شبکه عصبی با حافظه کوتاه مدت –بلند مدت
ورودی‌ها و خروجی‌ها	ثابت	متفاوت	متفاوت
معماری مدل	استفاده از فیلتر و شبکه عصبی پیش خور[9]	یک شبکه تکراری که شبکه با یافته‌ها تغذیه می‌شود.	داده‌های با اهمیت را برای مدت زمان طولانی در خود نگه می‌دارد.
موارد عملی	تشخیص چهره، تحلیل پزشکی، تحلیل تصویر، تشخیص و طبقه‌بندی تصاویر	ترجمه متن،تحلیل احساسات،پردازش زبان طبیعی، تحلیل گفتار	تحلیل گفتار،تشخیص و پردازش دست خط، تشخیص و پردازش تصویر
سناریوهای استفاده مناسب	داده‌های فضایی مانند تصویر	تحلیل داده‌های موقتی و متوالی مانند متن و ویدئو	تحلیل داده‌های موقتی و متوالی مانند متن و ویدئو

علاوه بر انواع شبکه‌های عصبی موجود، تحقیقات در حال انجام منجر به اکتشاف شبکه‌های جدید می‌شود. شبکه‌های عصبی را می‌توان بر اساس ساختار، جریان داده، گره‌های پردازشی (نرون‌ها)، چگالی، لایه‌ها و فیلترهای فعال‌سازی عمق طبقه‌بندی کرد [۲۵]. برخی از انواع شبکه‌های عصبی عمیق رایج شبکه‌های عصبی بازگشتی ساده[10]، شبکه‌های عصبی کانولوشنال[11] و شبکه‌های حافظه کوتاه‌مدت بلند مدت[12] هستند [۲۷]. در جدول ۱ این سه نوع شبکه با هم مقایسه شده است.

هر نوع شبکه عصبی بسته به نیازهای داده ورودی/خروجی خاص، سناریوها و موارد استفاده، نقاط قوت و ضعف خاص خود را دارد. ترکیب صحیح دو یا چند نوع شبکه عصبی عمیق می‌تواند کارایی متفاوتی نسبت به استفاده از یک نوع شبکه را مهیا کند.

[1] Proactive
[2] Reactive
[3] Packet_in
[4] Flow Table
[5] Original Packet
[6] Deep Learning
[7] Machine Learning
[8] Representation
[9] Feed Forward (FF)
[10] Sample Recurrent Neural Networks (S_RNN)
[11] Convolutional neural network (CNN)
[12] Long short-term memory (LSTM)

۲-۴- حملات امنیتی در شبکه‌های مبتنی بر نرم‌افزار

با وجود قابلیت‌ها و عملکردهای متعدد شبکه مبتنی بر نرم‌افزار، امنیت همچنان یک نگرانی حیاتی است. کنترل‌کننده این شبکه‌ها که هسته مرکزی و مسئول مدیریت جریان داده است، بالاترین خطر خرابی تک نقطه‌ای را به همراه دارد. به خطر انداختن کنترل‌کننده می‌تواند عواقب شدیدی در کل شبکه داشته باشد. پیکربندی نادرست در کنترل‌کننده‌ها می‌تواند منجر به عواقب خطرناکی شود، زیرا قابلیت برنامه‌ریزی آن‌ها را در معرض حملات احتمالی قرار می‌دهد و اعتبار، امنیت و یکپارچگی شبکه را به خطر می‌اندازد.

پیاده‌سازی مکانیسم‌های نظارت، تحلیل و پاسخ امنیتی در این شبکه‌ها می‌تواند به افزایش امنیت شبکه کمک کند. توجه به این نکته مهم است که حملات سایبری که این شبکه‌ها را هدف قرار می‌دهند، می‌توانند در مقایسه با شبکه‌های سنتی عواقب مخرب‌تری داشته باشند [۳۱].

جدول ۲ نمای کلی از انواع مختلف حملات بالقوه‌ای که می‌توانند در شبکه مبتنی بر نرم‌افزار رخ دهند، را ارائه می‌دهد، که بر اساس سطوح تهدید در لایه‌های مختلف این شبکه‌ها طبقه‌بندی می‌شوند [۳۲]. هر لایه الزامات امنیتی خاص خود را دارد. عدم رعایت این الزامات، شبکه را در معرض تهدیدات و حملات امنیتی مختلف قرار می‌دهد. اگر پیوند ارتباطی بین سوئیچ‌ها و کنترل‌کننده‌ها به خطر بیفتد، تمام سطوح سیستم در برابر حملات سیل‌آسا آسیب‌پذیر می‌شوند.

جدول ۲. حملات بالقوه در شبکه‌های مبتنی بر نرم‌افزار[۱۰]

روندهای امنیتی در شبکه مبتنی بر نرم‌افزار		لایه داده	لایه کنترل‌کننده	لایه کاربرد
دسترسی غیر مجاز	برنامه غیرمجاز	×	✓	✓
	تسلط بر کنترل‌کننده	✓	✓	×
مسائل مربوط به پیکربندی	عدم پیاده‌سازی TLS	✓	✓	✓
	اجرای سیاست	✓	✓	✓
حملات انکار سرویس توزیع شده	حمله سیل آسا به کنترل‌کننده	✓	✓	×
	حمله سیل آسا به سوئیچ	✓	×	×
دستکاری داده	دستکاری جریان داده	✓	✓	×
نشت داده	مدیریت اعتبار	×	✓	×
	کشف خط مشی ترافیک	×	✓	×

انواع مختلفی از حملات مخرب می‌توانند امنیت شبکه‌های مبتنی بر نرم‌افزار را به خطر بیندازند، از جمله حملات انکار سرویس توزیع شده، حملات جعل[۱]، شناسایی[۲] بسته‌هاو حدس زدن گذرواژه‌ها [۳۲].در این پژوهش روشی برای تشخیص حملات انکار سرویس توزیع‌شده ارائه می‌شود.

۳- کارهای مرتبط

در این بخش علاوه بر اینکه رویکردهای مبتنی بر یادگیری ماشین و یادگیری عمیق موجود برای تشخیص حملات انکار سرویس در کنترل‌کننده شبکه مبتنی بر نرم‌افزار مورد بررسی قرار می‌گیرد، روش ماشین بردار پشتیبان به عنوان یک روش یادگیری ماشین برای انتخاب ویژگی نیز تشریح می‌شود .

۳-۱- روشهای مبتنی بر یادگیری ماشین

رویکردهای مبتنی بر یادگیری ماشین با استفاده از الگوریتم‌های یادگیری ماشین توسعه یافته و بهبود می‌یابند. این رویکردها سیستم‌های تشخیص نفوذ[۳] را آموزش می‌دهند تا با تجزیه و تحلیل ویژگی‌های ترافیک شبکه، حملات انکار سرویس توزیع شده مخرب را شناسایی کنند. به عنوان مثال، روشی برای شناسایی و کاهش شدت این حملات در شبکه مبتنی بر نرم‌افزارتوسط خشاب و همکاران پیشنهاد شد [۳۳]. آنها از شش الگوریتم یادگیری ماشین شامل جنگل تصادفی[۴]، رگرسیون لجستیک[۵]، بیز[۶]، نزدیکترین همسایه[۷]، ماشین بردار پشتیبانی و درختان تصمیم[۸] استفاده کردند. نتایج روش پیشنهادی نشان داد که جنگل تصادفی از سایر الگوریتم‌ها بهتر عمل می‌کند و مناسب‌ترین طبقه‌بندی‌کننده برای روش آنها است.با این حال، نویسندگان جزئیات کافی در مورد مجموعه داده مورد استفاده یا اینکه حملات خاص انکار سرویس توزیع شده در نظر گرفته شده است یا نه ارائه نکردند[۱۰].

سودار و همکاران [۳۴] تحقیقی در مورد استفاده از ماشین بردار پشتیبانی و درخت تصمیم برای شناسایی حملات انکار سرویس توزیع شده در شبکه‌های مبتنی بر نرم‌افزار انجام داد. آنها رویکرد پیشنهادی خود را با استفاده از مجموعه داده استاندارد ارزیابی کردند. با این حال، روش آنها به عملکرد مناسبی دست نیافت و درختان تصمیم تنها به ۷۸ دست یافت. سانتوس و همکاران [۳۵] از پرسپترون چندلایه، الگوریتم جنگل تصادفی، ماشین بردار

[1] Spoofing
[2] Sniffing
[3] Intrusion Detection System (IDS)
[4] Random Forest(RF)
[5] Logistic Regression
[6] Naïve Bayes
[7] K-Nearest Neighbors (K-NN)
[8] Decision Tree

پشتیبان و درخت تصمیم برای شناسایی انواع مختلف حملات انکار سرویس توزیع شده، از جمله حملات نقطه‌ای، حملات جدول جریان سوئیچ، حملات کنترل‌کننده، و حملات پهنای باند استفاده کرد. آنها رویکرد خود را با استفاده از یک مجموعه داده واقعی ارزیابی کردند. با این حال، نتایج نشان‌دهنده عملکرد ضعیف طبقه‌بندی برای پرسپترون چندلایه و ماشین بردار پشتیبان در تشخیص حملات به کنترل‌کننده، با نرخ دقت تنها ۹۰ درصد شد.

در مقابل، دیپا و همکاران. [۳۷] یک الگوریتم ترکیبی یادگیری ماشین پیشنهاد کرد که ماشین بردار پشتیبانی و نقشه خود سازماندهی[1] (خوشه‌بندی) را برای شناسایی حملات انکار سرویس توزیع شده در شبکه‌های مبتنی بر نرم‌افزار ترکیب می‌کند. مدل ترکیبی به نرخ تشخیص[2] ۹۰,۴۵ درصد، نرخ دقیق تشخیص[3] ۹۶,۷۷ درصد و نرخ هشدار نادرست[4] ۰,۰۳۲ درصد دست یافت.

با این حال، رویکرد ترکیبی پیشنهادی دقت و عملکرد نرخ تشخیص پایین را نشان داد. علاوه بر این، نویسندگان اطلاعات محدودی در مورد مجموعه داده و نوع حمله انکار سرویس توزیع شده مدنظرشان ارائه کردند.

ماشین بردار پشتیبان

ماشین بردار پشتیبان[۲۷] یک الگوریتم یادگیری ماشین محبوب به دلیل عملکرد برتر آن است. در این مقاله از این روش برای انتخاب ویژگی استفاده می‌شود.

ماشین بردار پشتیبان برای طبقه‌بندی، یک مرز تصمیم (جداسازی ابرصفحه[5]) می‌سازد که نمونه‌های مثبت را از نمونه‌های منفی با حداکثر حاشیه جدا می‌کند. ابرصفحه جداکننده با یک تابع خطی $w \cdot x + b = 0$ نشان داده می‌شود، که در آن $w \in R^m$ بردار وزن و $b \in R$ برای جابجایی ابر صفحه جداکننده است. ایده بردار ماشین بردار پشتیبان را می‌توان به عنوان مسئله بهینه‌سازی محدود به صورت فرمول (۱) در نظر گرفت:

$$\min_{w,b,\xi} \frac{1}{2} \| w \|_2^2 + C \sum_{i=1}^{n} \xi_i \qquad (۱)$$

$$subject\ to\ y_i(w \cdot x_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0, i = 1, \ldots, n$$

که در آن $\| w \|$ نرم اقلیدسی را نشان می‌دهد. در اینجا، هدف عبارت اول، منظم‌سازی است که مربوط به اندازه‌گیری معکوس حاشیه جداکننده است، عبارت دوم خطای آموزشی را اندازه‌گیری می‌کند و ابرپارامتر $C > 0$ مبادله بین آنها را مشخص می‌کند.

الگوریتم ماشین بردار پشتیبان در ابتدا برای طبقه‌بندی باینری طراحی شد. درصورتی که، بسیاری از وظایف طبقه‌بندی دنیای واقعی شامل طبقه‌بندی چند کلاسه است. دو رویکرد رایج برای گسترش ماشین بردار پشتیبان برای شناسایی چندین کلاس وجود دارد. یک رویکرد این است که با استفاده از یک استراتژی اکتشافی، مانند استراتژی یک در مقابل یک[6]، مسئله را به چندین مسئله طبقه‌بندی باینری تقسیم کنیم. با استفاده از این رویکرد می‌توان چندین مدل ساخت. کلاس ورودی جدید با رای اکثریت تعیین می‌شود، یعنی کلاسی که بیشترین رای را داشته باشد به عنوان کلاس پیش بینی شده انتخاب می‌شود. هر مدل بردار ضریب متفاوتی را تحویل می‌دهد. در نتیجه، این رویکرد برای انتخاب ویژگی مناسب نیست[۲۸].

روش دیگری برای تعمیم الگوریتم ماشین بردار پشتیبان برای مسائل طبقه‌بندی چند کلاسه، رویکرد همه با هم [7]نامیده می‌شود[۲۹]. این رویکرد به طور همزمان چند ابرصفحه جداکننده را ایجاد می‌کند، که منجر به یک مرز تصمیم‌گیری تکه‌ای می‌شود که مجددا هدف به حداکثر رساندن حاشیه است . [۳۰]در این مقاله از ماشین بردار پشتیبان به عنوان روش مناسب و هوشمند برای انتخاب ویژگی بسته‌های شبکه استفاده شده است.

۳-۲- روش‌های مبتنی بر یادگیری عمیق

در سال‌های اخیر، الگوریتم‌های یادگیری عمیق نقش مهمی در سیستم‌های تشخیص نفوذ برای شناسایی حملات انکار سرویس توزیع شده در شبکه مبتنی بر نرم‌افزار ایفا می‌کند. الگوریتم‌های یادگیری عمیق ثابت کرده‌اند که بسیار کارآمد و مؤثر هستند و از رویکردهای مبتنی بر یادگیری ماشین پیشی گرفته‌اند. علاوه بر این، الگوریتم‌های یادگیری عمیق دارای قابلیت‌های قوی در تقلید از مغز انسان هستند که به آن‌ها اجازه می‌دهد ویژگی‌ها را به دست آورند و به طور خودکار ساختارهای عمیق را از داده‌های خام بیاموزند. چندین رویکرد مبتنی بر یادگیری عمیق پیشنهاد شده است.

به عنوان مثال، آلانازی و همکاران. [۳۸] یک رویکرد ترکیبی از واحد بازگشتی دروازه‌ای[8]، شبکه عصبی کانولوشنال و حافظه کوتاه‌مدت بلند مدت برای شناسایی حملات انکار سرویس توزیع

[1] Self-Organizing Map(SOM)
[2] Detection rate
[3] Accurate detection rate
[4] False alarm rate (FAR)
[5] Separating hyperplane
[6] One-versus-one(OVR)
[7] All-together
[8] Gated Recurrent Unit (GRU)

شده در شبکه‌های مبتنی بر نرم‌افزار پیشنهاد کرد. این روش با استفاده از مجموعه داده استاندارد مورد ارزیابی قرار گرفت و تنها با انتخاب چهار ویژگی به دقت تشخیص بالایی دست یافت.

سلسووا و همکاران [۳۶] روشی را پیشنهاد کرد که از یک شبکه عصبی عمیق[1] برای محافظت از صفحات داده و کنترل در مقابل حملات انکار سرویس در شبکه‌های مبتنی بر نرم‌افزار استفاده می‌کند. با این حال، آنها از مجموعه داده‌ای که به طور خاص برای محیط این شبکه‌ها طراحی نشده است، برای آموزش، آزمایش و ارزیابی سیستم پیشنهادی خود استفاده کردند. در نتیجه، این روش از نظر معیارهای محاسبه به عملکرد پایینی دست یافت.

حسیه و همکاران [۳۹] یک تکنیک یادگیری عمیق مبتنی بر شبکه عصبی کانولوشنال برای تشخیص حملات سیل آسا پیشنهاد کرد. آنها رویکرد خود را با استفاده از یک مجموعه داده تولید شده آزمایش کردند و نتایج را با نرخ دقت تشخیص ۹۵،۰۳٪ به دست آوردند. علاوه بر این، این رویکرد هزینه‌های سربار را در سطح کنترل‌کننده افزایش داد.

وان و همکاران [۴۰] از یک شبکه حافظه کوتاه مدت دو طرفه برای شناسایی چنین حملاتی در شبکه مبتنی بر نرم‌افزار استفاده کرد. روش پیشنهادی به تشخیص موفقیت آمیز با دقت بالا و نرخ‌های مثبت کاذب پایین دست یافت. با این حال، این رویکرد با استفاده از یک مجموعه داده غیر مرتبط آزمایش شد.

لی و همکاران [۴۱] یک سیستم تشخیص نفوذ بر اساس چهار الگوریتم یادگیری عمیق طراحی کرد: پرسپترون چند لایه، شبکه حافظه کوتاه مدت دو طرفه، رمزگذار خودکار پشته‌ای[2] و شبکه عصبی کانولوشنال، برای شناسایی حملات انکار سرویس توزیع شده و پوسته ایمن[3] در مقابل تلاش‌های همه جانبه در یک شبکه مبتنی بر نرم‌افزار استفاده کرد. پرسپترون چند لایه برای حملات انکار سرویس و پوسته ایمن برای حملات همه جانبه به ترتیب به دقت تشخیص بالایی ۹۸،۳ و ۹۹ درصد دست یافت. با این حال، عملکرد سایر الگوریتم‌های یادگیری عمیق در شناسایی چنین حملاتی به دلیل کمبود اطلاعات در مورد مجموعه داده‌ها و ویژگی‌ها ضعیف بود.

بوکریا و همکاران [۴۲] از الگوریتم‌های استاندارد یادگیری عمیق برای شناسایی تلاش‌ها برای حملات در کانال ارتباطی استفاده کرد که کنترل‌کننده را با صفحه زیرساخت پیوند می‌دهد. نویسندگان ادعا کردند که سیستم پیشنهادی به دقت تشخیص ۹۹،۶ درصد دست یافته است. با این حال، این رویکرد با استفاده از یک مجموعه داده غیر مرتبط که نشان دهنده یک شبکه مبتنی بر نرم‌افزار نیست، آزمایش و ارزیابی شد. علاوه بر این، فقط به محافظت از کانال‌های ارتباطی محدود می‌شود.

منصور و همکاران [۱۰] روشی را پیشنهاد می‌کنند که پس از انتخاب ویژگی‌های برجسته ازطریق مکانیزم‌های خاص، با استفاده از مدل شبکه‌های عصبی بازگشتی، حملات انکار سرویس توزیع شده در شبکه مبتنی بر نرم‌افزار را به طور موثر تشخیص می‌دهد و معیارهای ارزیابی میانگین دقت تشخیص[4]، میانگین دقت[5]، میانگین نرخ هشدارغلط [6] و میانگین هارمونیک[7] برای این روش به ترتیب برابر ۹۴،۱۸۶٪، ۹۲،۱۴۶٪، ۸،۱۱۴٪ و ۹۴،۲۷۶٪ بیان شده است.

انواع مختلفی از حملات از جمله حملات انکار سرویس توزیع شده، حملات جعل، شناسایی بسته‌ها، و حدس زدن گذرواژه‌ها یا حملات همه جانبه مخرب می‌توانند امنیت شبکه‌های مبتنی بر نرم‌افزار را به خطر بیندازند [۳۲]. این تحقیق مکانیزمی را برای انتخاب ویژگی ارائه می‌کند که علاوه بر این که مبتنی بر روش ماشین بردار پشتیبان است، به شناسایی حملات انکار سرویس توزیع شده کمک می‌کند. همچنین یک رویکرد مبتنی بر یادگیری عمیق از طریق ترکیب شبکه‌های عصبی بازگشتی ساده و حافظه کوتاه‌مدت بلند مدت با شبکه عصبی کانولوشن را معرفی می‌کند که در ادامه مورد بحث قرار خواهد گرفت.

۴- روش پیشنهادی

روش پیشنهادی این مقاله، یک روش مبتنی بر یادگیری عمیق برای تشخیص حملات انکار سرویس توزیع شده بر روی کنترل‌کننده شبکه مبتنی بر نرم‌افزار از طریق ویژگی‌های است. این روش، بسته‌های حمله را از بسته‌های معمولی ترافیک شبکه تفکیک می‌کند. این روش از سه مرحله کلی تشکیل شده است: در مرحله اول: پیش پردازش[8]، پاکسازی[9]، تبدیل[10]، نرمال‌سازی[11] و متعادل‌سازی[12] روی داده‌ها برای آماده شدن جهت ورود به مرحله دوم است. مرحل دوم: شامل استخراج ویژگی‌های برجسته در

[1] Deep Neural Network(DNN)
[2] Stacked Auto Encoder (SAE)
[3] Secure shell(SSH)
[4] Detection Accuracy(DA)
[5] Precision
[6] False Alarm rate(FAR)
[7] F1_measure
[8] Preprocessing
[9] Clean
[10] Transformation
[11] Normalization
[12] Balancing

تفکیک بسته‌های حاوی حملات و بسته‌های ترافیک معمولی شبکه صورت می‌پذیرد و در نهایت مرحله سوم: تشخیص بسته‌های مخرب از بسته‌ای عادی را با استفاده از مدل ترکیبی سه شبکه عصبی کانولوشنال، بازگشتی ساده و با حافظه کوتاه مدت و بلندمدت می‌باشد.

۴-۱- پیش پردازش

مجموعه داده استفاده شده در آموزش و آزمایش مدل پیشنهادی دارای مشخصات ذکر شده در جدول ۳ می‌باشد [۴۴]. پاکسازی داده، به فرایند تصحیح یا حذف داده‌های تکراری، معیوب، در فرمت نادرست یا ناقص اشاره دارد. تبدیل، فرایند تبدیل قالب داده به قالب یا ساختار دیگر است. برای مثال تبدیل داده رشته‌ای به داده عددی یا به طور کلی قابل بهره برداری برای الگوریتم‌های یادگیری عمیق.

نرمال‌سازی یکی از روش‌های بی مقیاس کردن داده است و از فرمول (۲) انجام می‌شود.

$$X_{SCALE} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (۲)$$

متعادل کردن داده‌ها، برای اطمینان از استفاده مساوی از هردو نوع عادی و حمله داده‌ها در شبیه‌سازی صورت می‌پذیرد. در این مقاله از روش افزایش نمونه[1] اسموت[2] برای افزایش تعداد نمونه حملات استفاده می‌شود.

۴-۲- انتخاب ویژگی

هدف اصلی از انتخاب ویژگی، تعیین ویژگی‌های با اهمیت و تاثیر گذار از میان همه ویژگی‌های بسته‌های شبکه، قبل از ورود به مدل شبکه عصبی، به منظور کاهش افزونگی، کاهش نویز و کاهش ابعاد نهایتا بهبود عملکرد کلی مدل در تشخیص حملات انکار سرویس توزیع شده است[۱۰]. معیار انتخاب ویژگی‌ها، بالا بودن قدرت پیش بینی است. روش‌های مختلفی و گاها ترکیبی از چند روش برای انتخاب ویژگی‌های تاثیر گذار در تشخیص حملات انکار سرویس توزیع شده وجود دارد .در این مقاله از روش ماشین بردار پشتیبان برای انتخاب ویژگی استفاده می‌شود. این روش در بخش۳,۱,۱ تشریح شد.

۴-۳- تشخیص حملات انکار سرویس توزیع شده

از ترکیب سه نوع شبکه عصبی کانولوشن، شبکه عصبی با حافظه کوتاه مدت، بلند مدت و شبکه عصبی بازگشتی ساده برای ساخت یک مدل تشخیص پیشنهاد می‌شود . این مدل می‌تواند به طور موثری حملات انکار سرویس توزیع شده بر روی کنترل‌کننده‌های شبکه مبتنی بر نرم‌افزار را با استفاده از ویژگی‌های تعیین‌کننده شناسایی کند. جدول ۴ معماری این شبکه عصبی ترکیبی را نشان می‌دهد.

جدول ۳. مجموعه داده جهت آموزش و آزمایش مدل پیشنهادی[۴۴]

مشخصات	جزئیات
تعداد کل رکورد	۱۰۴۳۴۵
تعداد رکورد حمله	۴۰۷۸۴
تعداد رکورد نرمال	۶۳۵۶۱
نوع بسته	نرمال و حمله
کلاس‌های نرمال	ICMP,UDP,TCP
کلاس‌های حمله	ICMP,UDP Flooding , TCP Syn
ویژگی‌های محاسبه شده	- ویژگی Pktrate تعداد بسته‌های ارسال شده در ثانیه است - ویژگی pktperflow تعداد بسته‌ها در یک جریان واحد است - ویژگیTot_durکل جریان ورودی به سوئیچ است - ویژگی packetins تعداد پیام‌های packetins است. - ویژگی Port_no جمع دو ویژگی tx_kbps و rx_kbps - ویژگی byteperflow ازشمارش بایتها درهرجریان به دست می‌آید. - ویژگی‌های tx_kbps و rx_kbps نشان دهنده نرخ ارسال و دریافت داده است.
ویژگی‌های اخذ شده از سوییچ‌ها	- ویژگیهای Switch-id و Packet count و Byte_count و Duration_secو Duration_nsec که برحسب نانو ثانیه است - ویژگی rx_byte تعداد بایت دریافت شده در پورت سوئیچ است. - ویژگی dt نشان دهنده تاریخ و زمان است که به عدد تبدیل شده است. - ویژگی tx_byte تعداد بایت ارسال شده از پورت سوئیچ است. - ویژگی‌های source IPو Destination IP و port number که به ترتیب آدرس مبدا و مقصد و شماره پورت را مشخص می‌کنند. - ویژگی Porotocol نوع پروتکل را مشخص می‌کند.

۵- پیاده‌سازی و آزمایش

در این بخش نتایج تجربی حاصل از پیاده‌سازی مدل پیشنهادی تشریح می‌شود.

[1] Oversampling

[2] SMOTE

جدول ۴. معماری شبکه ترکیبی پیشنهادی

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 3, 32)	128
max_pooling1d_1 (MaxPooling1D)	(None, 1, 32)	0
dropout_3 (Dropout)	(None, 1, 32)	0
lstm_1 (LSTM)	(None, 1, 32)	8320
simple_rnn_1 (SimpleRNN)	(None, 32)	2080
batch_normalization_1 (BatchNormalization)	(None, 32)	128
dropout_4 (Dropout)	(None, 32)	0
flatten_1 (Flatten)	(None, 32)	0
dense_2 (Dense)	(None, 64)	2112
dropout_5 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 2)	130

Total params: 12898 (50.38 KB)
Trainable params: 12834 (50.13 KB)
Non-trainable params: 64 (256.00 Byte)

۵-۱- معیارهای ارزیابی

اثر بخشی روش پیشنهادی با اندازه‌گیری دقت تشخیص، نرخ مثبت کاذب، حساسیت و میانگین هارمونیک بررسی می‌شود. این معیارها با استفاده از ماتریس درهم ریختگی[1] که در جدول ۵ تشریح شده، محاسبه می‌شود.

جدول ۵. ماتریس درهم ریختگی

کلاس پیش بینی	نرمال	حمله
واقعیت	منفی کاذب	مثبت صحیح
	منفی صحیح	مثبت کاذب

مثبت صحیح [2] یعنی طبقه‌بندی‌کننده حمله را به طور دقیق شناسایی کرده است. مثبت کاذب[3] به این معنی است که طبقه‌بندی‌کننده یک بسته عادی را به اشتباها به عنوان حمله طبقه‌بندی کرده است. منفی صحیح[4] یعنی طبقه‌بندی‌کننده به طور صحیح یک بسته عادی را برچسب زده است. منفی کاذب[5] مواردی را نشان می‌دهد که یک حمله به اشتباه توسط طبقه‌بندی‌کننده به عنوان بسته عادی برچسب خورده است

همچنین محققان در مطالعات خود سایر معیارهای ارزیابی را به صورت زیر تعریف کرده‌اند[۱۰]: فرمول (۳) نرخ هشدار اشتباه را که نشان‌دهنده تعداد بسته‌های نرمالی که به اشتباه حمله تشخیص داده شده‌اند را نسبت به کل بسته‌های نرمال نشان می‌دهد. هرچه این عدد پایین‌تر باشد اشتباه مدل در تشخیص بسته‌های نرمال به عنوان بسته حمله کمتر و نشان‌دهنده عملکرد بهتر مدل است.

فرمول(۴) دقت مدل را نشان می‌دهد که بیانگر نسبت تعداد بسته‌های حمله درست تشخیص داده شده به کل بسته‌هایی است که حمله تشخیص داده شده است. فرمول (۵) بیانگر میانگین هارمونیک دقت مدل است که از طریق فرمول (۶) و فرمول حساسیت به دست می‌آید. فرمول (۷) صحت مدل را نشان می‌دهد نسبت تشخیص صحیح مجموع حملات و ترافیک نرمال را به کل بسته‌ها نشان می‌دهد.

$$FAR = \frac{FP}{TN+FP} \quad (۳)$$

$$Precision = \frac{TP}{TP+FP} \quad (۴)$$

$$F1\ Measure = 2 \times \left(\frac{Precision \times Recall}{Precision+Recall}\right) \quad (۵)$$

$$Recall = \frac{TP}{TP+FN} \quad (۶)$$

$$DA = \frac{TP+TN}{TP+TN+FP+FN} \quad (۷)$$

۵-۲- مکانیزم انتخاب ویژگی

استفاده از ویژگی‌های مرتبط، برای آموزش مدل، می‌تواند تاثیر قابل توجهی در اثر بخشی رویکرد داشته باشد. در نتیجه ارتباط ۱۹ ویژگی شامل ۱۸ ویژگی به عنوان متغیر مستقل و ۱ ویژگی متغیر وابسته و هدف برای ورودی به الگوریتم ماشین بردار پشتیبان انتخاب می‌شود، رتبه‌بندی ویژگی‌ها بر اساس وزن ارتباط ویژگی‌های مستقل با ویژگی هدف[6] از طریق ماژول وزن ویژگی[7] نرم‌افزار رپیدماینر[8] در نمودار ۱ مشخص شده است. از میان ۱۸ ویژگی ۵ ویژگی که بیشترین وزن را دارند، به عنوان ورودی برای آموزش مدل پیشنهادی انتخاب می‌شوند.

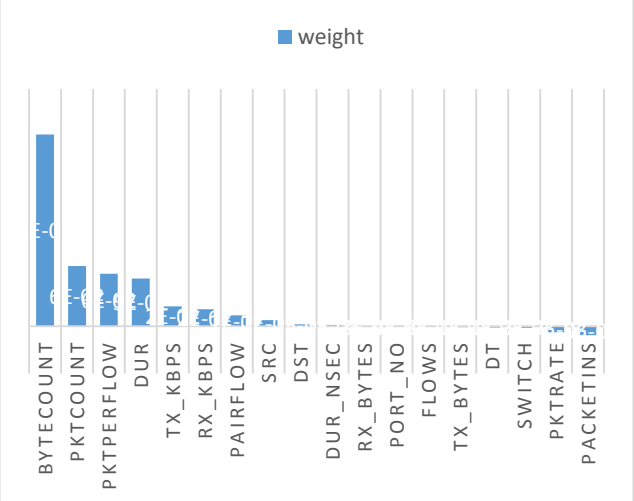
۵-۳- پیاده‌سازی مدل پیشنهادی

در این بخش، اطلاعات دقیقی در مورد پیکربندی مدل یادگیری و معماری آن ارائه می‌شود. مدل پیشنهادی با استفاده از ویژگی‌های انتخاب شده آموزش داده می‌شود.

این مدل ترکیبی از سه شبکه کانولوشنال، بازگشتی ساده و حافظه کوتاه مدت و بلندمدت می‌باشد که با نرخ یادگیری ۰،۰۱ و در ۱۰۰ تکرار برای یادگیری با استفاده از ۸۰ درصد داده‌های مجموعه داده آماده‌سازی شده در مرحله پیش پردازش همبندی شد. ماتریس

[1] Confusion matrix
[2] True Positive(TP)
[3] False Positive(FP)
[4] True Negative(TN)
[5] False Negative(FN)
[6] Target
[7] Feature weight
[8] RapidMiner

درهم ریختگی آزمایش مدل پیشنهادی با ۲۰ درصد باقیمانده از مجموعه داده در پنج اجرا صورت پذیرفت. نتایج آزمایش به طور میانگین به شرح جدول ۶ می‌باشد. همچنین معیارهای ارزیابی این آزمایش در جدول ۷ بررسی می‌شود.



نمودار۱. وزن ویژگی‌های مجموعه داده

جدول ۶. ماتریس درهم ریختگی مدل پیشنهادی (میانگین ۵ اجرا)

کلاس پیش‌بینی	بسته نرمال	بسته حمله
واقعیت	۶۶۰	۷۶۷۴
	۱۲۱۹۳	۳۴۲

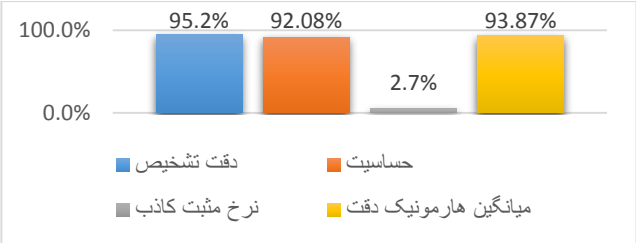
جدول ۷. معیارهای ارزیابی آزمایش میانگین ۵ اجرا(درصد)

دقت تشخیص	حساسیت	نرخ مثبت کاذب	میانگین هارمونیک دقت
۹۵,۲	۹۲,۰۸	۲,۷	۹۳,۸۷

۶- نتیجه‌گیری

نمودار۲ میانگین نتایج آزمایش مدل پیشنهادی را نمایش می‌دهد. در مدل پیشنهادی دقت تشخیص ۹۵,۲ درصد، حساسیت ۹۲,۰۸درصد، نرخ مثبت کاذب(نرخ هشدار اشتباه) ۲,۷درصد و میانگین هارمونیک دقت ۹۳,۸۷درصد را نشان می‌دهد.

جدول ۸ عملکرد موثر روش پیشنهادی در تشخیص حملات انکار سرویس توزیع شده را در مقایسه با روش‌های مبتنی بر یادگیری عمیق موجود نشان می‌دهد.



نمودار۲. میانگین نتایج آزمایش مدل پیشنهادی

جدول ۸. مقایسه معیارهای ارزیابی روش پیشنهادی با دیگر روش‌های مبتنی بر یادگیری عمیق موجود

	روش پیشنهادی	منصور و همکاران[۳]	حسیه و همکاران[۴۰]	بوکریا و همکاران[۴۳]
دقت تشخیص	٪۹۵,۲	٪۹۴,۱۸۶	٪۹۱,۹۷۶	٪۹۳,۲۰۳
حساسیت	۹۲,۰۸٪	٪۹۲,۱۴۶	٪۹۱,۷۷۲	٪۸۸,۴۷۲
نرخ مثبت کاذب	٪۲,۷	٪۸,۱۱۴	٪۸,۱۳۸	٪۱۲,۷۴۶
میانگین هارمونیک	٪۹۳,۸۷	٪۹۴,۲۷۸	٪۹۱,۹۳۲	٪۹۳,۵۴۶

در این جدول روش پیشنهادی با سه روش موجود دیگر مقایسه می‌شود. در معیار ارزیابی دقت تشخیص نسبت به سه روش دیگر، بهترین عملکرد را دارد، در معیار ارزیابی حساسیت رتبه دوم را دارد، در معیار نرخ خطای کاذب نسبت به سه روش دیگر درصد کمتر و درنتیجه عملکرد بهتری را نشان می‌دهد. در معیار میانگین هارمونیک دقت نیز رتبه دوم را دارد. لذا در مجموع عملکرد روش پیشنهادی نسبت به دیگر روش‌ها کارایی بالاتری را نشان می‌دهد.

آنچه باعث بهبود عملکرد مدل پیشنهادی نسبت به سایر مدل‌های مورد بررسی شده است، علاوه بر ترکیب مناسب سه نوع شبکه عصبی مطرح شده، استفاده از ماشین بردار پشتیبان به عنوان یک روش یادگیری ماشین برای انتخاب ویژگی به جای روش‌های آماری است.

۷- کارهای آینده

تهدیدات امنیتی باعث نگرانی در استفاده از شبکه مبتنی بر نرم‌افزار است. یکی از نقاط ضعف این شبکه‌ها کنترل‌کننده است. ایجاد یک مکانیزم مناسب برای تشخیص حملات انکار سرویس توزیع شده نقش زیادی در کاهش این نگرانی‌ها دارد. برای رسیدن به این هدف این مقاله یک رویکرد جدید مبتنی بر یادگیری عمیق پیشنهاد می‌کند. روش پیشنهادی شامل سه مرحله است و به طور موثر حملات انکار سرویس توزیع شده را تشخیص می‌دهد. این روش

نسبت به روش‌های موجود بهبود قابل توجهی در عملکرد نشان می‌دهد.

پژوهش در زمینه تشخیص حملات انکار سرویس توزیع شده در شبکه مبتنی بر نرم‌افزار موضوعات فراوانی برای کارهای آینده دارد. بررسی ترکیب انواع شبکه‌های عصبی عمیق، بخصوص انواع رمزگذارهای خودکار[1] و استفاده از دیگر روش‌های یادگیری ماشین برای انتخاب ویژگی می‌تواند منجر به افزایش قدرت مدل‌های تشخیص حمله در کنترل‌کننده‌های شبکه‌های مبتنی بر نرم‌افزار شود.

مراجع

[1] Muhammad, Tayyab. "Revolutionizing Network Control: Exploring the Landscape of Software-Defined Networking (SDN)." *International Journal of Computer Science and Technology* 3.1 pp 36-68, (2019).

[2] D. Kreutz et al., "Software-defined networking: A comprehensive survey," Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, 2015.

[3] S. Mousavi et al., "Early detection of DDoS attacks against SDN controllers," in Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC), pp. 77-81.

[4] M. A. Aladaileh et al., "Renyi Joint Entropy-Based Dynamic Threshold Approach to Detect DDoS Attacks against SDN Controller with Various Traffic Rates," Appl. Sci., vol. 12, no. 12, p. 6127, 2022.

[5] K. M. Sudar et al., "Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques," in Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-5.

[6] L. Wan, Q. Wang, and S. Zheng, "Deep SSAE-BiLSTM Model for DDoS Detection In SDN," in Proceedings of the 2nd International Conference on Computer Communication and Network Security (CCNS), pp. 1-4, 2021

[7] F. Alanazi et al., "Ensemble Deep Learning Models for Mitigating DDoS Attack in Software-Defined Network," Intell. Autom. Soft Comput, vol. 33, pp. 923–938, 2022.

[8] V. Deepa et al., "Detection of DDoS attack on SDN control plane using hybrid machine learning techniques," in Proceedings of the 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 299–303.

[9] R. Santos et al., "Machine learning algorithms to detect DDoS attacks in SDN," Concurr. Comput. Pract. Exp., vol. 32, p. e5402, 2020.

[10] A. Mansoor et al., "Deep Learning-Based Approach for Detecting DDoS Attack on Software-Defined Networking Controller," Systems, vol. 11, no. 6, p. 296, 2023.

[11] J. Karhunen, T. Raiko, and K. Cho, "Unsupervised deep learning: A short review," in Advances in Independent Component Analysis and Learning Machines, E. Bingham et al., Eds. Academic Press, 2015, pp. 125–142.

[12] A. K. Singh and S. Srivastava, "A survey and classification of controller placement problem in SDN," International Journal of Network Management, vol. 28, no. 3, p. e2018, 2018.

[13] K. Sood and Y. Xiang, "The controller placement problem or the controller selection problem?," Journal of Communications and Information Networks, vol. 2, no. 3, pp. 1-9, 2017.

[14] B. A. A. Nunes et al., "A survey of software-defined networking: Past, present, and future of programmable networks," IEEE Communications surveys & tutorials, vol. 16, no. 3, pp. 1617-1634, 2014.

[15] W. Xia et al., "A survey on software-defined networking," IEEE Communications Surveys & Tutorials, vol. 17, no. 1, pp. 27-51, 2014.

[16] Open Networking Foundation. [Online]. Available:https://www.opennetworking.org/about.

[17] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," IEEE Communications surveys & tutorials, vol. 16, no. 4, pp. 1955-1980, 2014.

[18] S. Sezer et al., "Are we ready for SDN? Implementation challenges for software-defined networks," IEEE Communications Magazine, vol. 51, no. 7, pp. 36-43, 2013.

[19] R. Trestian, K. Katrinis, and G. M. Muntean, "OFLoad: An OpenFlow-based dynamic load balancing strategy for datacenter networks," IEEE Transactions on Network and Service Management, vol. 14, no. 4, pp. 792-803, 2017.

[20] H. Wang, Y. Wang, and Y. J. Yan, "A distributed network traffic monitoring platform based on SDN," Electric Power Information and Communication Technology, vol. 14, no. 10, pp. 22-27, 2016

[21] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Netw., vol. 61, pp. 85–117, 2015.

[22] O. E. Elejla et al., "Deep-Learning-Based Approach to Detect ICMPv6 Flooding DDoS Attacks on IPv6 Networks," Appl. Sci., vol. 12, no. 12, p. 6150, 2022.

[23] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," Handb. Brain Theory Neural Netw., vol. 3361, 1995.

[24] X. Pan et al., "Recent methodology progress of deep learning for RNA–protein interaction prediction," Wiley Interdiscip. Rev. RNA, vol. 10, p. e1544, 2019.

[25] A. Dongare et al., "Introduction to artificial neural network," Int. J. Eng. Innov. Technol. (IJEIT), vol. 2, pp. 189–194, 2012.

[26] J. Karhunen et al., "Unsupervised deep learning: A short review," in Advances in Independent Component Analysis and Learning Machines, E. Bingham et al., Eds. Academic Press, 2015, pp. 125–142.

[27] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[28] F. Tang et al., "Group feature selection with multiclass support vector machine," Neurocomputing, vol. 317, pp. 42–49, 2018.

[29] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in Proceedings of the 7th European Symposium On Artificial Neural Networks, 1999, pp. 219–224.

[30] Y. Guo, Z. Zhang, and F. Tang, "Feature selection with kernelized multi-class support vector machine," Pattern Recognition, vol. 117, p. 107988, 2021.

[31] A. Akhunzada et al., "Securing software defined networks: Taxonomy, requirements, and open issues," IEEE Commun. Mag., vol. 53, pp. 36–44, 2015.

[32] A. Pradhan and R. Mathew, "Solutions to Vulnerabilities and Threats in Software Defined Networking (SDN)," Procedia Comput. Sci., vol. 171, pp. 2581–258, 2020.

[1] Auto Encoders (AE)

[33] Khashab, F.; Moubarak, J.; Feghali, A.; Bassil, C. DDoS attack detection and mitigation in SDN using machine learning. In Proceedings of the 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan, 28 June–2 July 2021; pp. 395–401

[34] K. M. Sudar et al., "Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques," in Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–5.

[35] R. Santos et al., "Machine learning algorithms to detect DDoS attacks in SDN," Concurr.Comput. Pract. Exp., vol. 32, p. e5402, 2020.

[36] B. Celesova et al., "Enhancing security of SDN focusing on control plane and data plane," in Proceedings of the 2019 7th International Symposium on Digital Forensics and Security (ISDFS), pp. 1–6.

[37] V. Deepa et al., "Detection of DDoS attack on SDN control plane using hybrid machine learning techniques," in Proceedings of the 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 299–303.

[38] F. Alanazi et al., "Ensemble Deep Learning Models for Mitigating DDoS Attack in Software-Defined Network," Intell. Autom. Soft Comput, vol. 33, pp. 923–938, 2022.

[39] C. Hsieh et al., "Efficient Detection of Link-Flooding Attacks with Deep Learning," Sustainability, vol. 13, p. 12514, 2021.

[40] L. Wan, Q. Wang, and S. Zheng, "Deep SSAE-BiLSTM Model for DDoS Detection In SDN," in Proceedings of the 2nd International Conference on Computer Communication and Network Security (CCNS), pp. 1–4, 2021

[41] T. H. Lee, L. H. Chang, and C. W. Syu, "Deep learning enabled intrusion detection and prevention system over SDN networks," in Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1–6.

[42] S. Boukria and M. Guerroumi, "Intrusion detection system for SDN network using deep learning approach," in Proceedings of the 2019 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS), Volume 1, pp. 1–6.

[43] M. Iqbal and M. Rizwan, "Application of 80/20 rule in software engineering Waterfall Model," in Proceedings of the 2009 International Conference on Information and Communication Technologies, pp. 223–228.

[44] https://www.kaggle.com/datasets/chiragchiku25/ddos-sdn-dataset