مجله‌ی علمی پژوهش در علوم رایانه، شماره ۳۸، تابستان ۱۴۰۴، ص ۲۸- ۱

ISSN: 2745-3673
http://www.Rcsj.ir

مجله علمی پژوهش در علوم رایانه (سال دهم)
شماره ۳۸ / تابستان ۱۴۰۴ / ص ۲۸- ۱

ارائه مدلی برای شناسایی نفوذ در شبکه اینترنت اشیا (*IOT*) با استفاده از الگوریتم‌های یادگیری عمیق و درخت تصمیم

جعفر پورامینی[۱]، جعفر عرفانی[۲]

[۱] عضو هیات علمی گروه مهندسی کامپیوتر و فناوری اطلاعات دانشگاه پیام نور، ایران.

[۲] کارشناسی‌ارشد مهندسی فناوری اطلاعات سیستم‌های تکنولوژی اطلاعات دانشگاه پیام‌نورمرکز بین‌المللی عسلویه، ایران.

نام نویسنده مسئول:
جعفر عرفانی
jafar.erfani30355@gmail.com

تاریخ دریافت: ۱۴۰۴/۰۳/۲۷
تاریخ پذیرش: : ۱۴۰۴/۰۴/۲۰

چکیده

فناوری اینترنت اشیا[۱] این قابلیت را به هر شیء هوشمندی می‌دهد که به شبکه اینترنت متصل شده و به ارسال و دریافت اطلاعات بپردازد، ترکیب این فناوری با سایر تکنولوژی‌ها مانند هوش مصنوعی[۲] باعث ایجاد سیستم‌ها و زیرساخت‌های هوشمندی شده است. در این شبکه حجم زیادی از داده تولید می‌شود و این می‌تواند یک فرصت یا تهدید باشد، بنابراین مهم‌ترین چالش اصلی، امنیت شبکه اینترنت اشیاست. اگرچه مکانیزم‌های امنیتی رایج تا حدی می‌توانند امنیت شبکه را تأمین کنند، اما توسعه روش‌ها و مدل‌های نفوذ، لزوم استفاده از راهکارهای جدید را دوچندان کرده است یکی از این روش‌ها استفاده از سیستم تشخیص نفوذ[۳] است که بر مبنای الگوریتم‌های یادگیری ماشین[۴] عمل می‌کند و می‌تواند ناهنجاری‌ها را در شبکه شناسایی کند. نرخ هشدار کاذب بالا و دقت نسبتاً کم از نقایص این سیستم‌های تشخیص نفوذ است؛ با پیشرفت در الگوریتم‌های شبکه‌های عصبی عمیق[۵] و توانایی این الگوریتم‌ها در مهندسی خودکار ویژگی‌ها، استفاده از این الگوریتم‌ها در جهت کاهش این معایب به عنوان یک نظریه مطرح شده است. بعلاوه در حوزه‌ای که داده‌های زیادی تولید می‌شود، ترکیب تکنیک‌های یادگیری ماشین با شبکه‌های عصبی عمیق، می‌تواند ضریب دقت تشخیص نفوذ را افزایش دهد. ایده اصلی این تحقیق ارائه مدلی برای افزایش دقت سیستم تشخیص نفوذ در شبکه اینترنت اشیا با استفاده از ترکیب الگوریتم شبکه عصبی کانولوشن[۶] و درخت تصمیم[۷] است. معماری سیستم تشخیص نفوذ پیشنهادی شامل دو مرحله است، در مرحله اول از بخش اول شبکه عصبی کانولوشن که شامل لایه‌های کانولوشن و پولینگ[۸] است، جهت استخراج ویژگی‌های بهینه استفاده شده، و در مرحله دوم معماری از درخت تصمیم که دقت نسبتاً خوبی در کلاس‌بندی داده‌ها دارد، برای دسته‌بندی ویژگی‌های خروجی از مرحله اول، استفاده شده است. در این پژوهش از

مجله‌ی علمی پژوهش در علوم رایانه، شماره ۳۸، تابستان ۱۴۰۴، ص ۲۸- ۱

ISSN: 2745-3673
http://www.Rcsj.ir

مجموعه‌داده NSL-KDD استفاده شده و جهت ارزیابی عملکرد از معیارهایی مانند دقت[۹]، صحت[۱۰] بازخوانی[۱۱] و نمره F1 استفاده شده است که به ترتیب مقادیر ۹۹/۷۳، ۹۹/۷۵، ۹۹/۷۵، و ۹۹/۷۴ درصد تجربه شده است که نسبت به مدل‌های پایه و هم ارز خود عملکرد بهتری را نشان داده است.

واژگان کلیدی: اینترنت اشیا، تشخیص نفوذ، یادگیری عمیق، درخت تصمیم.

مقدمه

اینترنت اشیا یا به اختصار IOT مفهوم نسبتاً جدیدی در دنیای فناوری اطلاعات محسوب می‌شود که به ارتباط اینترنتی بین اشیا و تجهیزاتی گفته می‌شود که در فضای پیرامونی اطراف ما قرار دارند. امروزه اتصال اشیا معمول مانند گوشی‌های هوشمند[۱]، رایانه‌ها، لپ‌تاپ‌ها[۲] و تبلت‌ها[۳] به اینترنت مزایای بسیاری برای پیشبرد کارهای بشری دارد، به طوری که چنین ابزارهایی بدون اینترنت تقریباً به یک شیء ناکارآمد تبدیل شده‌اند. در آینده نزدیک تقریباً هر شیء فیزیکی با استفاده از زیرساخت‌های شبکه‌ای و به لطف پیشرفت در صنعت ریزتراشه‌ها می‌تواند به شبکه اینترنت متصل شده و بدون دخالت مستقیم انسان به تبادل اطلاعات در شبکه بپردازد.

گسترش روزافزون اینترنت اشیا و اتصال میلیون‌ها دستگاه مختلف مانند سنسورها و اشیا هوشمند با پروتکل‌های ناهمگون و مختلف، تهدیدات و چالش‌های امنیتی را بیشتر کرده است و مکانیزم‌های امنیتی معمول مانند رمزنگاری[۱] و احراز هویت[۲] به تنهایی نمی‌توانند تضمین‌کننده امنیت در تمام سطوح آن باشد. یکی از مهم‌ترین این سازوکارهای امنیتی در حوزه شبکه، سیستم‌های تشخیص نفوذ هستند که با استفاده از الگوریتم‌های یادگیری ماشین با نظارت بر تمام ورودی‌های شبکه سعی در تشخیص نفوذ دارند. از طرفی چنین سیستم‌هایی به علت تولید حجم زیاد داده‌ها و رخ دادن حملات جدید و استفاده از الگوریتم‌های تک مدلی، دقت نسبتاً کم و بعضاً تشخیص‌های نادرستی دارند، از این نظر، جهت‌گیری پژوهش‌ها، به سمت سیستم‌های تشخیص نفوذ با استفاده از الگوریتم‌های یادگیری عمیق متمایل شده است که در برخورد با داده‌های حجیم بهتر عمل می‌کنند و می‌توانند با استخراج خودکار ویژگی‌ها، کارایی و عملکرد سیستم تشخیص نفوذ را بهبود دهند؛ نظر به ایرادات بیان شده سیستم‌های تشخیص نفوذ مبتنی بر یادگیری ماشین و از طرفی توسعه و کارایی بالای شبکه‌های عصبی که الهام گرفته از پردازش مغز انسان هستند و بنابراین هدف اصلی در این مقاله این است: آیا می‌توان با ترکیب الگوریتم‌های شبکه عصبی عمیق مانند کانولوشن که با افزایش داده‌ها عملکرد بهتری دارد و الگوریتم درخت تصمیم که در دسته‌بندی ویژگی‌ها دقت خوبی دارد، ضریب دقت سیستم تشخیص نفوذ را مقداری افزایش داد؟ تشخیص نفوذ با استفاده از ترکیب الگوریتم‌های یادگیری ماشین و عمیق، از مهم‌ترین مسائل مربوط به امنیت شبکه است و برای شناسایی انواع مختلفی از حملات که ممکن است توسط فایروال‌ها قابل‌شناسایی نباشند، توسعه و تکامل‌یافته‌اند و افزایش دقت این سیستم‌ها می‌تواند راهگشایی برای این چالش باشد.

اینترنت اشیا وسایل بسیاری را از طریق زیرساخت‌های سیمی یا بی‌سیم به هم متصل و شبکه‌ای را ایجاد کرده که اطلاعات را از منابع مختلف جمع‌آوری و با سرعت بالا با دستگاه‌های دیگر ارتباط برقرار می‌کنند و خدمات مختلف را تحت شبکهٔ واحد یکپارچه ارائه می‌دهد.

در این میان سیستم‌های تشخیص نفوذ اگرچه نمی‌توانند جایگزین مکانیسم‌های امنیتی فعلی شوند، اما در ترکیب با این راهکارها جهت شناسایی رفتارهای مخرب و ناهنجار، می‌تواند امنیت شبکه را بیشتر کنند. این سیستم‌ها با فیلترکردن شبکه و تجزیه‌وتحلیل داده‌های کاربر، الگوهای نفوذ را شناسایی و به مدیر شبکه جهت تصمیم مقتضی گزارش می‌دهد. در حال حاضر طیف گسترده‌ای از الگوریتم‌های هوش مصنوعی در سیستم‌های تشخیص نفوذ استفاده می‌شود؛ اما روش‌های ترکیبی با استفاده از شبکه‌های عصبی عمیق و یادگیری ماشین که از زیرشاخه‌های هوش مصنوعی هستند انعطاف‌پذیری بیشتری دارند و می‌توانند با بهبود کارایی و دقت، ضریب امنیت شبکه اینترنت اشیا را بیشتر کنند، در چنین روش‌هایی به دلیل استخراج خودکار

ISSN: 2745-3673
http://www.Rcsj.ir

ویژگی‌های، توانایی تشخیص نفوذ و تمایز بین الگوهای نرمال و نفوذ را در شبکه با دقت بیشتری انجام می‌شود و به طور خودکار نفوذها را از ترافیک عادی شبکه به صورت اعلان ناهنجار، شناسایی می‌کنند. با توجه به مسئله اصلی و ضرورت تحقیق، یک هدف کلی داریم که جهت رسیدن به این هدف کلی، تعدادی زیر هدف تعریف، و همین‌طور فرضیه‌هایی برای نیل به اهداف تعیین شده در نظر گرفته شده است. هدف ما در این مقاله افزایش دقت سیستم تشخیص نفوذ در شبکه‌های اینترنت اشیا با استفاده از الگوریتم‌های یادگیری عمیق و درخت تصمیم است.

مبانی نظری

اینترنت اشیا

اینترنت، شبکه‌ای جهانی است که از مجموعه کاربران و سرورها تشکیل شده است، و ساختار آن در حال دگرگونی است. بیش از یک دهه است که عضو جدیدی در قالب محصولات هوشمند در حال اضافه شدن به این شبکه بزرگ است که از آن به عنوان اشیا یاد می‌شود، این شیء می‌تواند هر دستگاهی باشد که دارای سنسور جهت تبادل اطلاعات است.

در یک نگاه کلی می‌توان گفت اینترنت اشیا شبکه‌ای از دستگاه‌های فیزیکی است که دارای قطعات الکترونیکی و سنسورها و اتصالات هوشمند و می‌توانند در بستر شبکه اینترنت، داده‌ها را تولید و ارسال کنند. در این فناوری به هر شیء یک شناسه منحصربه‌فرد تعلق می‌گیرد تا فرستنده و گیرنده دادها مشخص شوند. این فناوری به عنوان عضو جدایی‌ناپذیر از آینده شبکه اینترنت با استفاده از پروتکل‌های ارتباطی می‌تواند کامپیوترها، گوشی‌ها، تلویزیون‌ها، خودروها را در دایره خود قرار دهد، همین‌طور جدیدترین نوآوری مربوط به اینترنت اشیا می‌تواند همه قاره‌ها را به هم متصل کند و زندگی شخصی و اجتماعی انسان‌ها توسط این فناوری بهبود یافته و باعث تحولی شگرف در دنیای بشری خواهد شد.

کاربردهای اینترنت اشیا

با توجه به ارتباط اینترنت اشیا و هوشمندی وسایل که هدف اصلی آن افزایش بهره‌وری، کاهش زمان و هزینه است، اینترنت اشیا کاربردهای زیادی پیدا کرده است به‌طوری که هزینه تجهیزات استفاده شده در این شبکه در سال ۲۰۱۷ به ۲۷ میلیارد دلار رسید و پیش‌بینی می‌شود تا سال ۲۰۳۰ به ۱۲۵ میلیارد دلار برسد **[25]**. این فناوری در حوزه مختلفی کاربرد دارد که به تعدادی محدود و مهم از آنها اشاره می‌کنیم.

اینترنت اشیا در خانه‌ها و شهرهای هوشمند

شاید از مهم‌ترین کاربردهای اینترنت اشیا در زندگی آینده بشری، توسعه خانه‌های هوشمند و به تبع آن شهرهای هوشمند است. در یک‌خانهٔ هوشمند انواع وسایل الکترونیکی می‌توانند با استفاده از تکنولوژی‌ها شامل سنسورها، شبکه‌های اتصالی، دستگاه‌های کنترلی و نرم‌افزارهای کاربردی به هم متصل شده و از طریق شبکه اینترنت اشیا، امکان برقراری ارتباط و کنترل از راه دور اجزای مختلف خانه فراهم شود و از این طریق مدیریت وسایل کارآمدتر می‌شود. از امکانات خانه‌های هوشمند می‌توان به سیستم‌های روشنایی هوشمند، پایش و کنترل مصرف انرژی، سیستم‌های حفاظتی و امنیتی مانند نظارت بر دوربین‌های مداربسته، قفل‌های هوشمند و لوازم هوشمند، پایش و ثبت فعالیت‌های فیزیکی و سلامتی افراد، خودکارسازی فرایندهای روزمره و....اشاره کرد.

اینترنت اشیا در مراکز درمانی

استفاده از اینترنت اشیا، در مراکز درمانی و سیستم‌های بهداشتی می‌تواند به بهبود کیفیت خدمات پزشکی، افزایش بهره‌وری و بهبود عملکرد بهداشت و درمان کمک کند. از جمله کاربردهای ساده و اولیه اینترنت اشیا در مراکز درمانی و پزشکی می‌توان به پایش بیماران و تشخیص بیماری‌ها و مراقبت از راه دور بیماران بستری در منزل با استفاده از سنسورها و پوشیدنی‌های هوشمند[1]، سیستم‌های هشداردهنده موارد اورژانسی که می‌توانند ضربان قلب یا فشار خوان افراد بیمار را کنترل کرده و در صورت لزوم به پزشک معالج از طریق سامانه هوشمند اطلاع دهد را نام برد.

ISSN: 2745-3673
http://www.Rcsj.ir

معماری اینترنت اشیا

اگرچه فناوری اینترنت اشیا، امروزه به عنوان یکی از کاربردی‌ترین فناوری‌ها تبدیل شده، اما در حال حاضر هنوز استانداردی برای معماری آن به صورت جامع و کامل طراحی نشده است. با توجه به نحوه کارکرد و نوع پیاده‌سازی آن، معماری شبکه اینترنت اشیا نیز ممکن است متفاوت باشد، اما در یک ساختار کلی معماری اینترنت اشیا از چهارلایه تشکیل شده است**[10]** :

لایه فیزیکی: این لایه پایین‌ترین سطح در معماری است و شامل اشیایی که به سنسورها مجهز بوده و قابلیت دریافت داده‌ها از محیط پیرامونی خود را دارند.

لایه شبکه: تجهیزات لایه فیزیکی جهت ارسال داده‌ها، نیازمند واسطه‌هایی هستند که بتوانند داده‌ها را از منابع مختلف جمع‌آوری و جهت پردازش به لایه بالاتر بفرستد، این لایه به صورت شبکه‌های ارتباطی بی‌سیم یا سیمی و با استفاده از بلوتوث یا وای‌فای و پروتکل‌های مختلف تشکیل شده است.

لایه پردازش: این لایه به صورت یک میان‌افزار در یک فضای ابری عمل می‌کند و داده‌های دریافتی از شبکه را پردازش، ذخیره و در صورت نیاز تجزیه‌وتحلیل می‌کند. این لایه مانند پلی میان شبکه و لایه کاربرد عمل می‌کند و اطلاعات را پس از تجزیه‌وتحلیل به فرمت قابل‌استفاده لایه کاربرد تبدیل می‌کند.

لایه اپلیکیشن یا کاربرد: در این لایه اطلاعات به شکل قابل‌فهم و به فرمت‌های مختلف به مدیر یا کاربر نمایش داده می‌شود.

امنیت اینترنت اشیا

بر اساس مطالب گفته شده در مورد اینترنت اشیا، این فناوری اگر چه بسیار جذاب و کاربردی است و می‌تواند برای بشر آسایش به ارمغان بیاورد، اما یک وجه تاریک و چالش برانگیزی در بعد امنیت دارد. در ساختار و معماری اینترنت اشیا که باید اطلاعاتی را بعضاً در مقیاس کلان ارسال و به‌اشتراک‌گذاری قرار دهد، امکان نشت اطلاعات نیز وجود دارد. مدیریت آینده این شبکه، به دلیل تعداد زیاد دستگاه‌های مورداستفاده، با انواع خدمات، توپولوژی‌ها، برنامه‌ها و پروتکل‌ها دشوار است **[15]**. به همین دلیل پروتکل‌های یکپارچه‌سازی اینترنت در معرض خطرات و ضعف‌های امنیتی بالقوه‌ای قرار دارند که ممکن است امکان سرقت اطلاعات مربوط به داده‌های معمول را فراهم کند. بعلاوه در خصوص این فناوری خطراتی وجود دارد که نمی‌توان به طور صددرصد امنیت آن را تضمین کرد. خطرات ذاتی که در هر شبکه‌ای از جمله اینترنت اشیا که بر پایه اینترنت کار می‌کند وجود دارد، اینکه هر شیء که بتواند به اینترنت متصل شود به صورت بالقوه می‌تواند یک تهدید باشد، از طرفی با ساختار لایه‌ای اینترنت اشیا، در هر لایه‌ای چالش‌ها و نگرانی امنیتی خاص خود وجود دارد. در این میان امنیت در اینترنت اشیا، شامل طیف گسترده‌ای از اقدامات، استراتژی‌ها برای حفظ محرمانگی، حریم خصوصی و اعتماد میان کاربران است که هدف آن کاهش آسیب‌پذیری اشیا در برابر نفوذگران در سطوح مختلف است. مکانیزم‌های مختلفی وجود دارد که می‌توان برای افزایش امنیت در سیستم‌های کامپیوتری و شبکه‌ای اتخاذ کرد. سه سطح حفاظت در این مورد وجود دارد: **[18]**

پیشگیری از حمله: شامل فایروال‌ها

تشخیص حمله: سیستم‌های تشخیص نفوذ

واکنش به حمله: رمزگذاری

علی‌رغم اتخاذ مکانیسم‌های رمزنگاری و پروتکل‌ها جهت کنترل ارتباطات بین رایانه‌ها و کاربران، جلوگیری از همه نفوذان تقریباً غیرممکن است **[13]**

مفهوم نفوذ در اینترنت اشیا

نفوذ به معنای یک تهدید امنیتی و ورود غیرمجاز و دسترسی به دستگاه‌ها، سنسورها، مانند وسایل خانگی هوشمند و سایر اشیای متصل به اینترنت است. نفوذ به این دستگاه‌ها و سیستم‌ها معمولاً توسط افراد یا گروه‌هایی که قصد خرابکاری، دزدی اطلاعات یا دسترسی غیرمجاز به سیستم‌ها را دارند، انجام می‌شود.

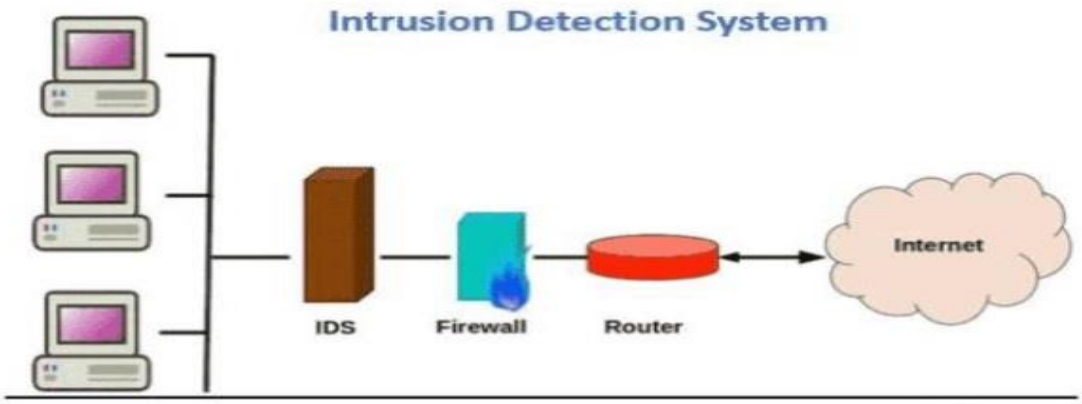
نفوذ به یک شبکه از مهم‌ترین آسیب‌های امنیتی در یک شبکه است، و به عملیاتی گفته می‌شود که به صورت غیرقانونی، صحت، محرمانگی و یا دسترسی به یک منبع را به خطر می‌اندازد. تشخیص نفوذ در شبکه اینترنت اشیا به معنای شناسایی

ISSN: 2745-3673
http://www.Rcsj.ir

فعالیت‌های مشکوک یا غیرمعمول شامل دسترسی‌های غیرمجاز به دستگاه‌ها، ارسال داده‌های ناخواسته و تلاش برای ورود به سیستم‌های متصل به شبکه باشد. برای تشخیص نفوذ در شبکه‌های اینترنت اشیا، ابزارها و فناوری‌های مختلفی وجود دارد که بر اساس الگوهای مشخص، فعالیت‌های ناهنجار را شناسایی و به مدیران سیستم اعلام می‌کنند.

هدف یک سیستم تشخیص نفوذ شناسایی موقعیت‌هایی است که در آن اقدامات کاربر با استفاده معمول از یک کامپیوتر یا شبکه کامپیوتری در تضاد است، مانند زمانی که آنها قصد کلاهبرداری دارند، به سیستم نفوذ می‌کنند تا داده‌ها را سرقت کنند، یا باعث اخلال در عملکرد یا خرابی سیستم می‌شود.

نمونه‌ای از سیستم تشخیص نفوذ را نشان می‌دهد که به صورت غیرفعال به یک سوئیچ متصل شده است. این سیستم می‌تواند بین فایروال و سوئیچ نصب شود تا تمام ترافیک ورودی به شبکه را اسکن کند.



شکل۱:سیستم تشخیص نفوذ در شبکه [19]

طبقه‌بندی سامانه‌های تشخیص نفوذ

سامانه‌های تشخیص نفوذ را می‌توان با توجه به روش‌های تشخیص و توسعه، به دو حالت کلی طبقه‌بندی کرد **[31]**. دسته‌بندی سیستم‌های تشخیص نفوذ نشان داده شده است.

سیستم‌های تشخیص نفوذ مبتنی بر روش تشخیص

این سیستم‌ها به نوبه خود به دودسته تقسیم می‌شوند: **[28]**

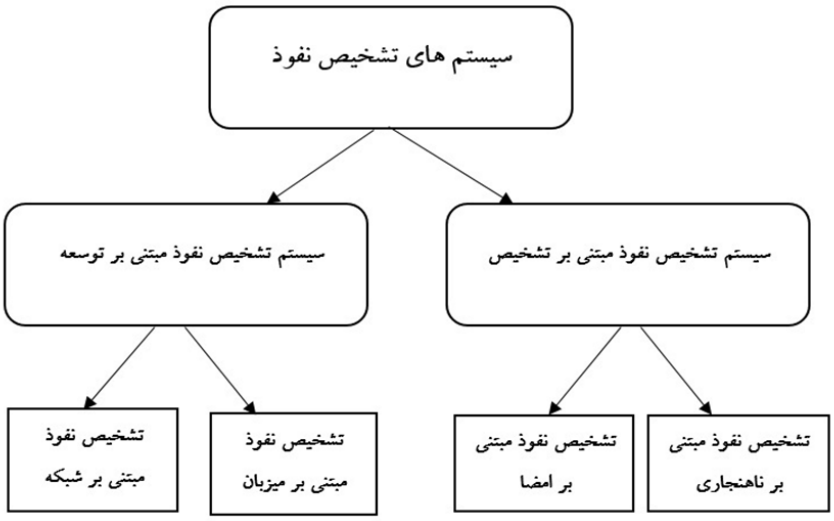
تشخیص نفوذ بر اساس ناهنجاری[1]

در این روش ابتدا با استفاده از روش‌های یادگیری ماشین یک الگویی از رفتارهای عادی ایجاد می‌شود، رفتارهایی که از این الگو پیروی کنند به عنوان رفتار عادی تلقی شده، و رویدادهایی که انحرافی غیرعادی از این الگو داشته باشند به صورت رفتار غیرعادی یا ناهنجار تشخیص داده می‌شود **[16]**

تشخیص نفوذ بر اساس امضا[2]

در این روش امضاها یا الگوهای از حملات پیش تعریف شده در پایگاه‌داده خود ذخیره و نگهداری می‌کند که هر کدام از این الگوها نوع خاصی از حمله را در خود ذخیره کرده، در صورت ایجاد چنین الگویی، بلافاصله از سوی سیستم تشخیص اعلام نفوذ می‌شود. مشکل اصلی این روش عدم شناخت حملات ناشناخته و جدید است.**[17]**

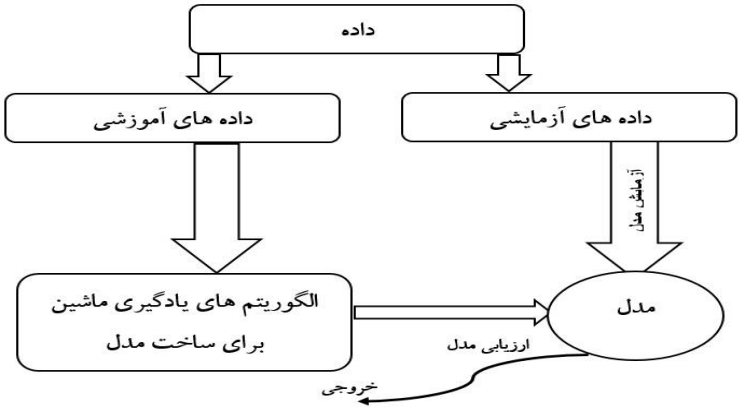
ISSN: 2745-3673
http://www.Rcsj.ir



شکل۲:دسته‌بندی سیستم‌های تشخیص نفوذ

یادگیری ماشین

یادگیری ماشین یکی از شاخه‌های هوش مصنوعی است که در آن ماشین وظایفی را که به‌صراحت برنامه‌ریزی نشده، یاد می‌گیرد و الگوهایی را برای پیش‌بینی با کمترین دخالت انسان شناسایی و عملکرد خود را با استفاده از تجربه‌ها، بهبود و تطبیق می‌دهد. این یادگیری در حال تبدیل‌شدن به یک فناوری مؤثر برای دنیای امروز و آینده به دلیل توانایی یادگیری از داده‌های برچسب‌دار و بدون برچسب است.

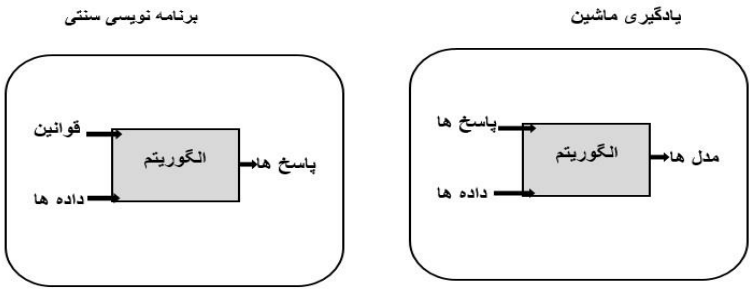


شکل۳:چارچوب کلی الگوریتم‌های یادگیری ماشین

تفاوت یادگیری ماشین و برنامه‌نویسی سنتی

یادگیری ماشین و برنامه‌نویسی سنتی از لحاظ عملکردی تفاوت‌هایی با هم دارند. در یادگیری ماشین الگوریتم‌ها به طور خودکار از داده‌ها یاد می‌گیرند و الگوهای پنهان در داده‌ها را با استفاده از آموزش تشخیص می‌دهند و توانایی پیش‌بینی و تصمیم‌گیری در مورد داده‌های جدید دارند، اما در برنامه‌نویسی، برنامه‌نویسان قوانین و الگوریتم‌های خاصی را به صورت دستی طراحی و پیاده‌سازی می‌کنند.

ISSN: 2745-3673
http://www.Rcsj.ir



شکل۴:تفاوت یادگیری ماشین و برنامه‌نویسی سنتی

رویکردهای اصلی یادگیری ماشین

سه رویکرد اصلی در یادگیری ماشین وجود دارد: یادگیری با نظارت[۴]، یادگیری بدون نظارت[۵] و یادگیری تقویتی[۶].**[9]**

یادگیری با نظارت

در یادگیری با نظارت که متداول‌ترین نوع یادگیری ماشین است، داده‌ها (ورودی‌ها) برچسب‌گذاری شده هستند؛ یعنی برای هر ورودی یک خروجی وجود دارد و مدل با استفاده این داده‌های برچسب زده شده آموزش می‌بیند و می‌تواند الگویی را بین داده‌ها و خروجی‌ها به صورت یک تابع یاد بگیرد و برای داده‌های جدید، خروجی مناسبی را پیش‌بینی کند.

این یادگیری بر اساس نوع داده‌ها به دو مسئله طبقه‌بندی[۱] (داده‌های گسسته) و رگرسیون[۲] (داده‌های پیوسته) تقسیم‌بندی می‌شود. وظیفه رگرسیون در مدل یادگیری، نگاشت داده‌ها به صورت یک معادله بین ورودی و خروجی است. در حالت طبقه‌بندی هر یک از داده‌های در یک دسته خاصی طبقه‌بندی می‌شوند.

یادگیری بدون نظارت

یادگیری بدون ناظر یکی از روش‌های یادگیری ماشین است که بدون نظارت انسان، می‌تواند الگوهای پنهان در داده‌ها را کشف کند. برای آموزش یادگیری بدون نظارت، داده‌ها فاقد برچسب‌گذاری هستند، به عبارتی ورودی‌ها مشخص هستند؛ اما خروجی‌ها معلوم نیستند و الگوریتم‌ها، الگوها و ساختار پنهان موجود در داده‌های بدون برچسب را پیدا می‌کنند. این دو یادگیری که بیشترین کاربرد در الگوریتم‌های یادگیری ماشین دارند تفاوت‌هایی دارند که در جدول زیر آورده شده است.

جدول ۱: تفاوت یادگیری با نظارت و بدون نظارت**[2]**

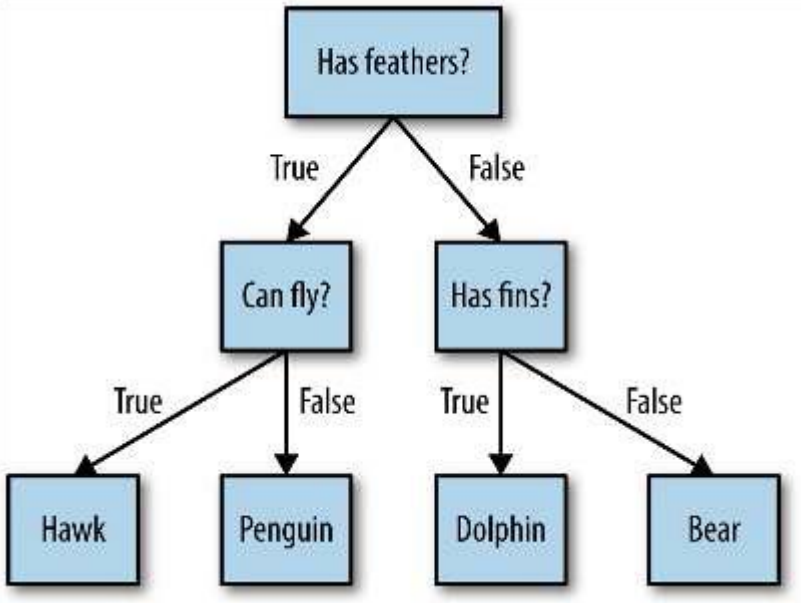
شاخص	یادگیری با نظارت	یادگیری بدون نظارت
تعریف	از طریق مجموعه‌داده دارای برچسب یاد می‌گیرد	بدون راهنما از طریق داده‌های بدون برچسب آموزش داده می‌شود
نوع داده‌ها	داده‌های دارای برچسب	داده‌های بدون برچسب
نوع مسئله	دسته‌بندی و رگرسیون	قوانین انجمنی و خوشه‌بندی
ناظر	ناظر اضافی	بدون ناظر
هدف	نگاشت داده‌های ورودی به خروجی‌های مشخص	کشف الگو

درخت تصمیم

یکی از الگوریتم‌های مهم یادگیری ماشین است که در حوزه یادگیری با نظارت قرار می‌گیرد، و با هر دو مسائل گسسته و پیوسته به‌خوبی کار می‌کند. درخت تصمیم نوعی مدل بدون پارامتری است که هم برای دسته‌بندی و هم برای رگرسیون

ISSN: 2745-3673
http://www.Rcsj.ir

استفاده می‌شوند. این الگوریتم به صورت یک ساختار درختی از سؤالات دودویی بر روی ویژگی داده‌ها ایجاد می‌شود تا بتواند داده‌ها را به دسته‌های مختلف تقسیم‌بندی کند و از یک گره ریشه، شاخه‌ها، گره‌های داخلی (مقادیر ویژگی‌ها) و گره‌های انتهایی یا همان برگ‌های درخت که برچسب کلاس هستند، تشکیل شده است. گره ریشه که در بالاترین سطح و نقطه شروع است بر اساس معیارهای بهینه‌سازی؛ مانند آنتروپی انتخاب می‌شود. با هر سؤالی که روی یک ویژگی پرسیده می‌شوند یک شاخه به عنوان نتیجه یا مقادیر آن ویژگی ایجاد می‌شود که ممکن است منجر به ایجاد یک پرسش روی ویژگی دیگر شود.



شکل۵:نمونه ساختار درخت تصمیم **[11]**

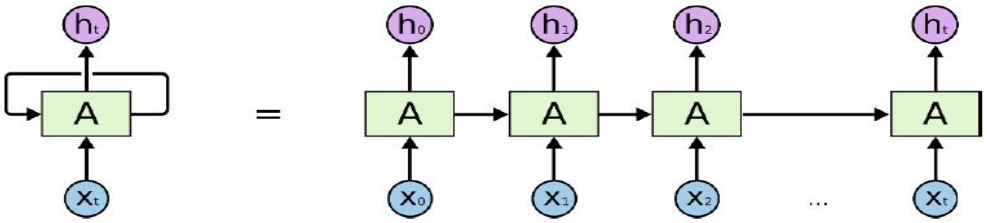
یادگیری عمیق

یادگیری عمیق که به آن شبکه‌های عصبی عمیق نیز می‌گویند زیرمجموعه‌ای از هوش مصنوعی و یادگیری ماشین است که بر استفاده از شبکه‌های عصبی مصنوعی برای حل مسائل پیچیده تمرکز دارد. به نوعی همان یادگیری ماشین است با تعداد لایه‌های مخفی و میانی زیاد که می‌تواند الگوهای پیچیده و پنهان را از داده‌ها استخراج کند. در مدل‌های یادگیری عمیق با پردازش داده‌ها بوسیله الگوگیری از مغز انسان، ویژگی بهینه و خاص را با استفاده از لایه‌هایی که در ساختارهای میانی خود دارند استخراج کرده و سعی در ایجاد یک الگویی هستند که می‌توان در حل مسائل از آن استفاده کرد. یادگیری عمیق لایه‌های مختلفی دارد که هر لایه ویژگی خاصی از ورودی را استخراج می‌کند. برخلاف یادگیری ماشین که ویژگی به صورت دستی استخراج می‌شود، در یادگیری عمیق استخراج ویژگی‌ها توسط لایه‌ها به صورت خودکار انجام می‌شود.

شبکه عصبی بازگشتی (RNN)

دسته‌ای دیگر از شبکه‌های عصبی است که از لایه‌های متصل به هم تشکیل شده و برای پردازش داده‌های سری زمانی و داده‌های ترتیبی مناسب است در این شبکه‌ها دو نوع ورودی داریم: خروجی مرحله قبلی و ورودی جدید و این یعنی نیاز به داشتن حافظه برای ثبت خروجی مرحله قبلی است. به این حافظه، وضعیت پنهان گفته می‌شود که با نماد **h** و ورودی اصلی با **x** نشان داده می‌شود. در (شکل ۲-۱۱) نمایی از شبکه عصبی بازگشتی نشان داد شده است. در هنگام ورود این دو به شبکه عصبی بازگشتی یعنی **ht-1** و **xt** در ماتریس‌ها وزنی خود ضرب و جمع شده یک بایاس هم اضافه، پس از عبور از یک تابع غیرخطی، خروجی **ht** را تولید می‌کند. این شبکه‌ها می‌توانند داده‌های دنباله‌دار را به خاطر داشتن حافظه، پردازش کنند. در شبکه‌های معمولی فرض بر این است که همه ورودی‌ها از همدیگر مستقل هستند، اما واحدهای شبکه عصبی بازگشتی با داشتن حافظه پنهان، آنچه رو که تاکنون دیده در خود جای می‌دهند و این ویژگی این امکان را می‌دهد تا الگوها را یاد بگیرند و در حافظه خود ذخیره کنند.

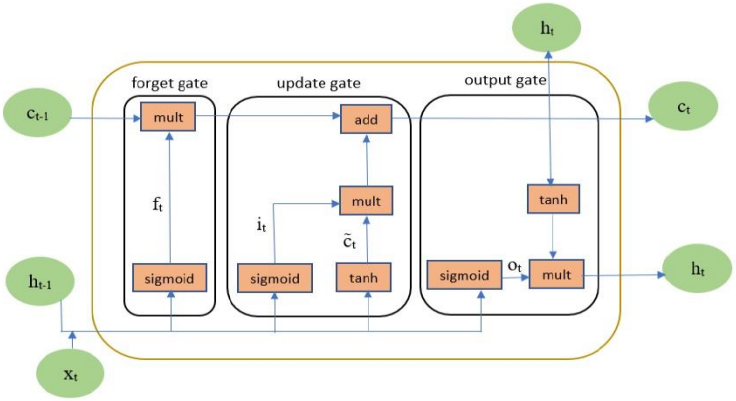
ISSN: 2745-3673
http://www.Rcsj.ir



شکل۶:شبکه عصبی بازگشتی **[26]**

شبکه عصبی حافظه طولانی کوتاه‌مدت (LSTM)

نوعی از شبکه‌های عصبی بازگشتی هستند که در جهت رفع ناپایداری گرادیان[۲] شبکه‌های بازگشتی و یادگیری وابستگی‌های بلندمدت طراحی شده‌اند و می‌توانند با داشتن حافظه‌ی بلند، داده‌هایی با طول بیشتری را یاد بگیرند. ساختار این شبکه تقریباً شبیه شبکه عصبی بازگشتی است، اما به جای دو ورودی، سه ورودی، x_t و h_{t-1} و c_{t-1} داریم که c_{t-1} حافظه سلول وضعیت است که مشخص می‌کند چه اطلاعاتی از توالی‌های گذشته می‌توانند در بلوک بعدی تأثیر داشته باشند؛



شکل۷:ساختار شبکه حافظه طولانی کوتاه‌مدت **[14]**

ادبیات پیشینه تحقیق

سیستم‌های تشخیص نفوذ مختلفی در پژوهش‌های خارجی و داخلی برای تشخیص نفوذ در شبکه‌های اینترنت اشیا ارائه شده، الگوریتم‌های یادگیری ماشین و اخیراً یادگیری عمیق و ترکیبی به طور گسترده‌ای در این سیستم‌ها استفاده شده‌اند.

جدول۲:خلاصه پیشینه تحقیق

خلاصه تحقیق	موضوع تحقیق	مرجع تحقیق
استفاده از الگوریتم درخت تصمیم برای انتخاب ویژگی‌ها و گسسته‌سازی ویژگی‌های پیوسته در بخش پیش‌پردازش	تشخیص نفوذ بلادرنگ با استفاده از ترکیب گسسته‌سازی و انتخاب ویژگی‌های	رحیم طاهری و همکاران ۲۰۱۷
با استفاده از الگوریتم k-maens و شبکه بیزین ویژگی‌ها انتخاب شده و شناسایی حملات توسط درخت تصمیم	تشخیص نفوذ در شبکه‌های رایانه‌ای با استفاده از درخت تصمیم و کاهش ویژگی‌ها	علی‌اکبر تجری سیاه مرز کوه ۲۰۲۱

ISSN: 2745-3673
http://www.Rcsj.ir

خلاصه تحقیق	موضوع تحقیق	مرجع تحقیق
با استفاده از الگوریتم‌های (MMI)، (GA)و آزمون f ویژگی‌ها به صورت اشتراکی انتخاب شده، و تنظیم فراپارامترها در الگوریتم ترکیبی عمیق دسته‌بندی CNN-LSTM توسط الگوریتم‌های گرگ خاکستری و نهنگ	شبکه عصبی عمیق ترکیبی بهینه ادغام شده با انتخاب ویژگی برای سامانه تشخیص نفوذ در حملات سایبری	مظلوم و بیگدلی ۲۰۲۳
جهت انتخاب پویای فراپارامترهای شبکه عصبی ALEXNET از الگوریتم میگوی آشوبی استفاده شده	بهبود تشخیص نفوذ به شبکه اینترنت اشیا با استفاده از یادگیری عمیق و الگوریتم بهینه‌سازی میگوی آشوبی	زنده‌دل و حمید زاده ۲۰۲۳
انتخاب ویژگی‌های بهینه با استفاده از الگوریتم گرگ خاکستری و ارزیابی ویژگی توسط جنگل تصادفی	ارائه یک الگوی فراابتکاری تشخیص نفوذ به کمک انتخاب ویژگی مبتنی بر گرگ خاکستری بهبودیافته و جنگل تصادفی	محمدی و همکاران ۲۰۲۳
در حالت طبقه‌بندی دودویی و دسته‌بندی چند کلاسی دقت نسبتاً بالایی داشته و در مقایسه با روش‌های سنتی ماشین دقت و نرخ تشخیص بالاتری داشته	یک رویکرد یادگیری عمیق برای تشخیص نفوذ با استفاده از شبکه‌های عصبی بازگشتی	[35]
شبکه عصبی عمیق با استفاده از یک چارچوب مقیاس‌پذیر جهت شناسایی و طبقه‌بندی حملات سایبری پیش‌بینی‌نشده	رویکرد یادگیری عمیق برای سیستم تشخیص نفوذ هوشمند	[34]
یک روش تجزیه‌وتحلیل برای شناسایی رفتارهای عادی و ناهنجار با استفاده یادگیری ماشین	یک سیستم تشخیص نفوذ مبتنی بر ناهنجاری برای وای‌فای	[27]
یک رویکرد دومرحله‌ای، مرحله اول برای تشخیص جریان نفوذ به‌وسیله درخت دودویی، مرحله دوم دسته‌بندی کلاس نفوذ با ترکیب (ET، RF و DNN)	رویکرد دومرحله‌ای برای تشخیص و شناسایی نفوذ در محیط‌های محاسباتی اینترنت اشیا و مه	[29]
استخراج ویژگی‌ها با استفاده از الگوریتم cnn ، سپس وارد مدل ترکیبی تشخیص نفوذ RWO شده که ترکیبی از الگوریتم بهینه‌سازی Remora (ROA) و الگوریتم بهینه‌سازی نهنگ (WOA) است.	یادگیری عمیق ترکیبی مبتنی بر بهینه‌سازی نهنگ Remora برای تشخیص نفوذ شبکه با استفاده از ویژگی‌های CNN	[30]
از الگوریتم شبکه عصبی کانولوشنال برای استخراج ویژگی، از درخت تصمیم برای دسته‌بندی ویژگی در کلاس‌ها	سیستم تشخیص نفوذ ترکیبی برای شبکه‌های اینترنت اشیا بی‌سیم با استفاده از الگوریتم یادگیری عمیق	[31]
ابتدا از الگوریتم کانولوشنال برای به‌دست‌آوردن ویژگی‌های بهینه، سپس یک تکنیک انتخاب ویژگی کارآمد بر	رویکرد تشخیص نفوذ برای محیط‌های ابری و اینترنت اشیا با استفاده از یادگیری عمیق و الگوریتم جستجوی کاپوچینو	[33]

ISSN: 2745-3673
http://www.Rcsj.ir

خلاصه تحقیق	موضوع تحقیق	مرجع تحقیق
اساس یک بهینه‌ساز توسعه‌یافته به نام الگوریتم جستجوی کاپوچینو (CapSA) استفاده است		

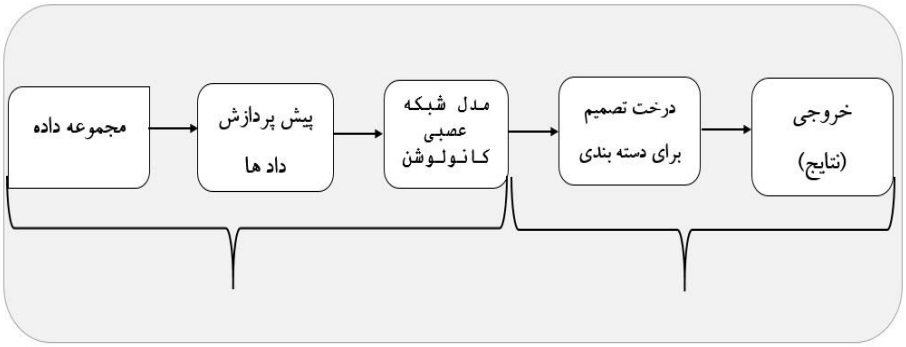
تجزیه و تحلیل داده‌ها:

نظر به اهمیت موضوع امنیت و حریم خصوصی و قابلیت اعتماد و اطمینان در شبکه‌های اینترنتی از جمله اینترنت اشیا که در حال فراگیرشدن در تمامی ابعاد آن از جمله تجارت فناوری اطلاعات و سایر حوزه ها، بعلاوه موضوع اصلی این مقاله نیز امنیت است، روش‌های مختلفی برای محافظت از داده‌ها و اطلاعات در شبکه وجود دارد، یکی از این راهکارها، سیستم‌های تشخیص و مقابله با نفوذ است که جهت تشخیص و مقابله با حملاتی از جمله نفوذ، انتشار ویروس‌ها استفاده می‌شود. سیستم پیشنهادی تشخیص نفوذ در شبکه اینترنت اشیا، مبتنی بر الگوریتم های یادگیری ماشین و عمیق است که ترکیبی از شبکه عصبی کانولوشنی و درخت تصمیم است.

کلیات روش کار به این صورت است که ابتدا مجموعه‌داده مربوط به سیستم‌های تشخیص نفوذ، پیش‌پردازش شده، سپس داده‌ها جهت استخراج ویژگی‌ها وارد شبکه عصبی کانولوشن شده‌اند و در نهایت ویژگی‌های استخراجی جهت دسته‌بندی به دو کلاس نفوذ و نرمال وارد بخش طبقه‌بندی‌شده‌اند. تازگی کار پژوهشی نوع ترکیب‌بندی معماری مدل تشخیص نفوذ در بخش انتهایی (دسته بندی) شبکه عصبی کانولوشن است. در این مدل به‌جای یک شبکه عصبی کاملاً متصل، از الگوریتم درخت تصمیم به عنوان طبقه‌بندی‌کننده در معماری مدل پیشنهادی استفاده شده است. در بخش آخر مدل یادگیری عمیق شبکه عصبی کانولوشن از یک لایه ی کاملاً متصل به صورت پیش‌خور برای طبقه‌بندی ویژگی‌ها استفاده می‌شود که باعث ازدست‌رفتن همسایگی می‌شود و برای بهینگی به پارامترهای بیشتری نیاز دارد؛ بنابراین جهت استفاده از ویژگی‌های خروجی بخش اول شبکه عصبی کانولوشن به شیوه کارآمد از درخت تصمیم برای دسته‌بندی ویژگی‌ها در مدل تشخیص نفوذ استفاده شده است. با توجه به سادگی و کارآمدی درخت تصمیم در مدل‌های یادگیری ماشین، تفسیر آن ساده و همین‌طور به پاک‌سازی داده‌های کمتری نیاز دارند.

مجموعه‌داده

بر اساس معماری نشان‌داده‌شده سیستم تشخیص نفوذ ابتدا باید داده‌هایی برای این معماری در نظر گرفت. در این پروژه از یکی از جدیدترین دیتاست ها به نام **NSL-KDD** که مرتبط با سیستم‌های تشخیص نفوذ است و دارای ۴۳ ستون که ۴۲ ستون مربوط به ویژگی‌ها و ۱ ستون مربوط به برچسب داده‌هاست، استفاده شده است. در ستون برچسب یعنی خروجی دیتاست، ۲۲ نوع حمله که به نوبه خود در زیر کلاس‌های **Dos**[۱]، **Probe**[۲]، **U2R**[۳]، **R2L**[۴] دسته بندی می شوند و یک کلاس **Normal** هم برای نمونه های عادی وجود دارد.

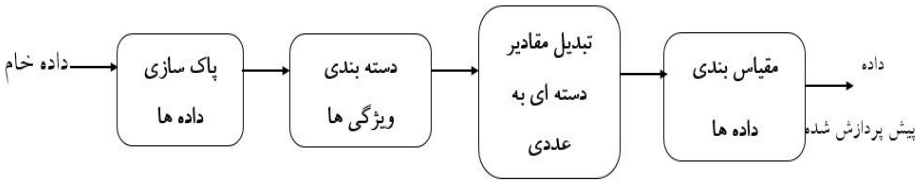


شکل۸: معماری سیستم تشخیص نفوذ ترکیبی

ISSN: 2745-3673
http://www.Rcsj.ir

مرحله پیش‌پردازش داده‌ها

داده‌ها برای ورود به الگوریتم باید به صورت یک ساختار مناسب در قالب جدول با رکوردها در سطرها و ویژگی‌ها در ستون‌ها باشند. ازآنجاکه داده‌ها در قالب مناسب نیستند ابتدا آنها را به صورت فایل **CSV** که فرمت قابل‌قبول الگوریتم‌های یادگیری ماشین است درآورده شده است. (شکل ۳-۲) فرایند پیش‌پردازش داده‌ها را نشان می‌دهد.



شکل۹:فرایند پیش‌پردازش داده‌ها

در اولین خطوط برنامه کتابخانه‌های لازم برای انجام پروژه مانند **Numpy** ، **Pandas** ، **TensorFlow** ، **Keras** توسط کدهای زیر وارد برنامه شده است.

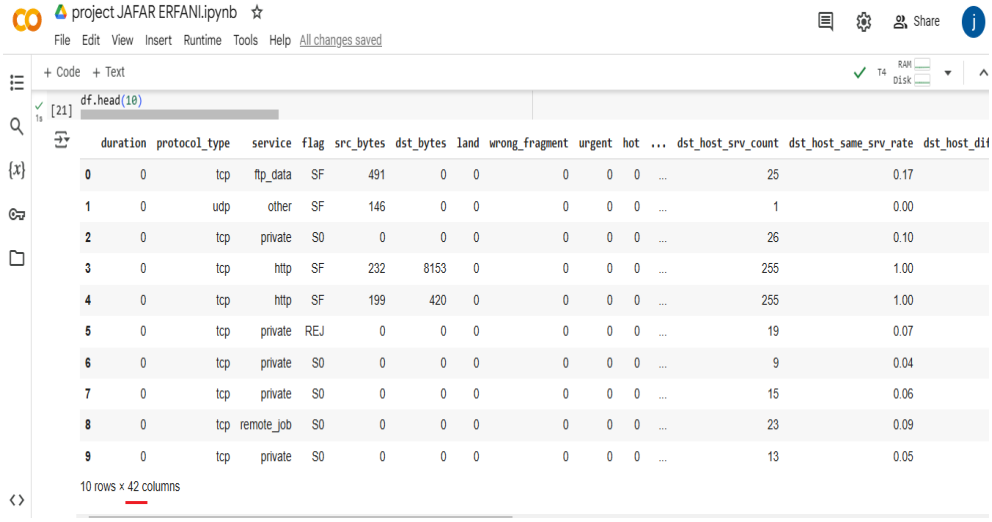
```
# Import necessary libraries
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers,models
from keras.models import Sequential,Model
from keras.layers import, Dropout, Flatten, ,Conv1D ,MaxPooling1D
from sklearn.preprocessing import  LabelEncoder, MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,
recall_score,f1_score,precision_score,confusion_matrix,classification_re
port
```

برای ورود داده‌ها به برنامه ابتدا داده‌ها که به فرمت مناسب جدولی **CSV** درآورده شده با استفاده از دستور زیر خوانده و داده‌ها به صورت یک دیتافریم در متغیر **df** قرار گرفته است.

```
df=pd.read_csv("nsl-kdd.csv")
```

بخشی از دیتاست با ۴۲ ویژگی به صورت زیر است.

ISSN: 2745-3673
http://www.Rcsj.ir



project JAFAR ERFANI.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

T4 RAM Disk

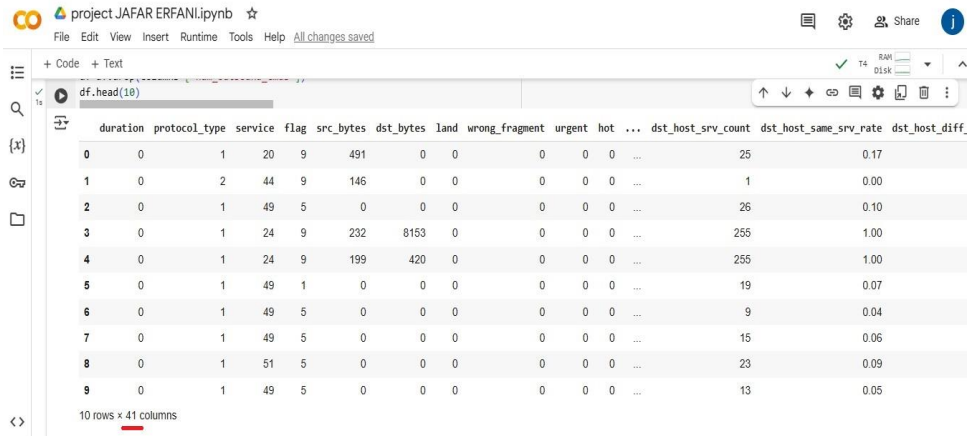
[21] df.head(10)

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_dif
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25	0.17	
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00	
2	0	tcp	private	S0	0	0	0	0	0	0	...	26	0.10	
3	0	tcp	http	SF	232	8153	0	0	0	0	...	255	1.00	
4	0	tcp	http	SF	199	420	0	0	0	0	...	255	1.00	
5	0	tcp	private	REJ	0	0	0	0	0	0	...	19	0.07	
6	0	tcp	private	S0	0	0	0	0	0	0	...	9	0.04	
7	0	tcp	private	S0	0	0	0	0	0	0	...	15	0.06	
8	0	tcp	remote_job	S0	0	0	0	0	0	0	...	23	0.09	
9	0	tcp	private	S0	0	0	0	0	0	0	...	13	0.05	

10 rows × 42 columns

ستون مربوط "**num_outbound_cmds**" دارای مقادیر صفر است، یعنی این ویژگی تأثیری در نتایج پروژه نخواهد داشت برای همین در اولین مرحله از پیش‌پردازش، این ستون را که دارای مقادیر **null** است حذف شده است.

نتیجه بعد از حذف ستون گفته شده به صورت زیر است:



project JAFAR ERFANI.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

T4 RAM Disk

df.head(10)

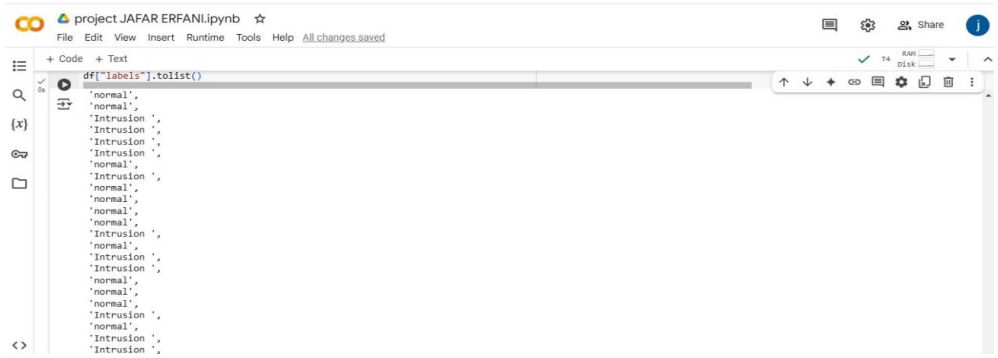
	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_
0	0	1	20	9	491	0	0	0	0	0	...	25	0.17	
1	0	2	44	9	146	0	0	0	0	0	...	1	0.00	
2	0	1	49	5	0	0	0	0	0	0	...	26	0.10	
3	0	1	24	9	232	8153	0	0	0	0	...	255	1.00	
4	0	1	24	9	199	420	0	0	0	0	...	255	1.00	
5	0	1	49	1	0	0	0	0	0	0	...	19	0.07	
6	0	1	49	5	0	0	0	0	0	0	...	9	0.04	
7	0	1	49	5	0	0	0	0	0	0	...	15	0.06	
8	0	1	51	5	0	0	0	0	0	0	...	23	0.09	
9	0	1	49	5	0	0	0	0	0	0	...	13	0.05	

10 rows × 41 columns

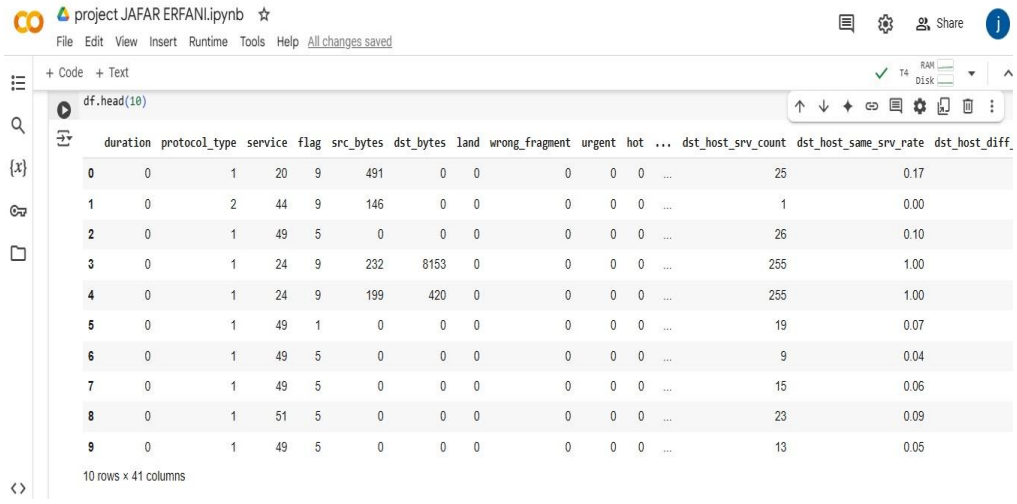
تعداد ستون‌ها یکی کم شده است.

در این پروژه، مهم‌ترین کار در مرحله پیش‌پردازش، دسته‌بندی مقادیر ویژگی برچسب یعنی خروجی به دو کلاس نفوذ و نرمال جهت افزایش دقت خروجی مدل تشخیص نفوذ است؛ بنابراین در این پژوهش مقادیر برچسب داده‌ها یعنی خروجی‌ها بر خلاف روش‌های معمول که به چهار کلاس حمله و یک کلاس عادی دسته‌بندی می‌کنند، به دو کلاس دسته‌بندی شده است. چون هدف، افزایش دقت تشخیص نفوذ است نه دسته‌بندی نوع حمله. مقادیر برچسب مربوط به حمله همه رکوردها را با واژه **Intrusion** ، و سایر مقادیر برچسب که مربوط به رکوردهای طبیعی و سالم بودند با کلمه **Normal** جایگزین شده است. خروجی مقادیر برچسب داده‌ها پس از تغییرات اعمال شده به صورت زیر است:

ISSN: 2745-3673
http://www.Rcsj.ir



در مرحله بعدی از مراحل آماده‌سازی دادها، همان‌طور که در خروجی قبلی مشاهده می‌شود تعدادی از ستون‌ها دارای مقادیر غیرعددی هستند که باید جهت ورود به الگوریتم تشخیص نفوذ به صورت مقادیر عددی باشند، با استفاده از تابع **LabelEncoder()** ستون‌های غیرعددی **protocol_type**، **service**، **flag** که مقادیر غیرعددی دارند به مقادیر عددی تبدیل شده است.خروجی این مرحله پس از تغییر مقادیر ویژگی های غیرعددی به عددی به صورت زیر است:



همان‌طور که ملاحظه می‌شود مقادیر ستون‌ها به عددی تغییر یافته‌اند.

در ادامه کار از مراحل پیش‌پردازش داده‌ها، چون مقیاس داده‌ها یکنواخت نیستند باعث ایجاد مشکلاتی در پردازش الگوریتم ها می‌شود، زیرا الگوریتم‌های یادگیری ماشین فقط مقادیر عددی را متوجه می‌شوند و چنانچه تفاوت زیادی در محدوده اعداد باشد این فرض برای الگوریتم ایجاد می‌کند که اعداد بالاتر برتری‌هایی دارند و می‌توانند نقش تعیین‌کننده‌ای در جهت‌گیری مدل ایفا کنند.

برای مقیاس‌بندی داده‌ها از تابع نرمال‌سازی[1] حداقل - حداکثر **MinMaxScaler()** جهت یکدست شدن داده‌ها استفاده شده که تمامی مقادیر ستون‌ها را در محدوده ۰ و ۱ مقیاس‌بندی می‌کند.

تابع نرمال‌سازی به صورت زیر است:

رابطه (۳-۱)

$$y = \frac{y - min}{max - min}$$

سپس دیتافریم[2] را با قراردادن مقادیر آن در متغیر **df_arry** به صورت یک آرایه[3] در می‌آوریم تا بتواند توسط الگوریتم های شبکه عصبی پردازش شوند. در ادامه مجموعه‌داده‌ها را به ویژگی‌های ورودی (**x**) و ویژگی برچسب (**y**) تقسیم می‌کنیم. با انجام این کار ۴۰ ستون اول را به عنوان ویژگی‌های ورودی در متغیر **x** و ستون آخر را در متغیر **y** قرار داده شده است.

ISSN: 2745-3673
http://www.Rcsj.ir

خروجی داده‌ها پس از مقیاس‌بندی به صورت زیر است:

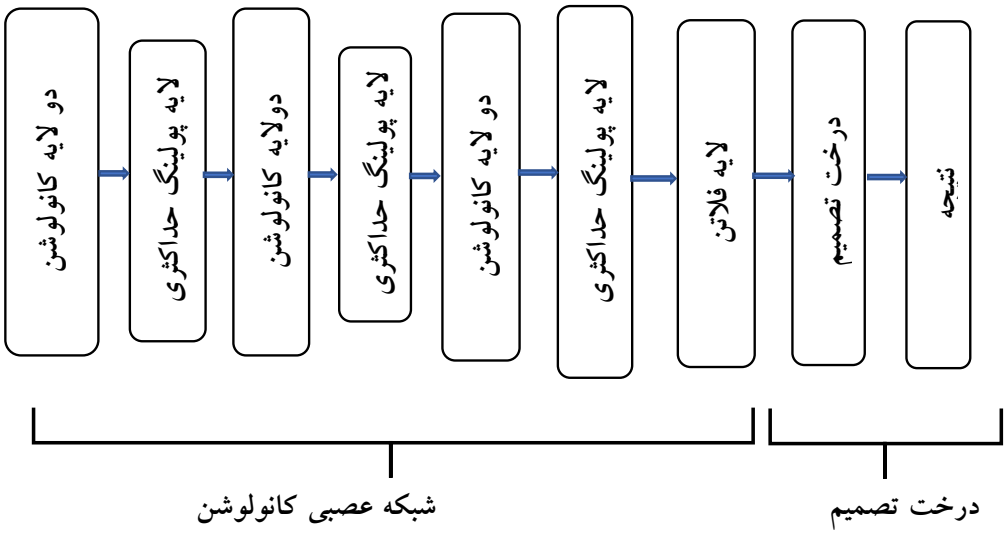
```
x_scale

(40,)
array([[0.5       , 0.28985507, 0.9       , ..., 0.        , 0.05      ,
        0.        ],
       [1.        , 0.63768116, 0.9       , ..., 0.        , 0.        ,
        0.        ],
       [0.5       , 0.71014493, 0.5       , ..., 1.        , 0.        ,
        0.        ],
       ...,
       [0.5       , 0.7826087 , 0.9       , ..., 0.        , 0.01      ,
        0.        ],
       [0.5       , 0.43478261, 0.5       , ..., 1.        , 0.        ,
        0.        ],
       [0.5       , 0.28985507, 0.9       , ..., 0.        , 0.        ,
        0.        ]])
```

مشاهده می‌شود که خروجی داده‌ها پس از انجام مقیاس‌بندی در محدوده صفر و یک قرار دارند.
خروجی داده‌ها دوبعدی هستند و برای ورود به مدل شبکه عصبی کانولوشن نیاز به تغییر ابعاد دارند
داده‌های پیش‌پردازش شده در این مرحله جهت ورود به شبکه عصبی کانولوشن تغییر ابعاد می‌دهند.

```
x_scale_reshaped=x_scale.reshape(x_scale.shape [0] ,x_scale.
shape[1],1)
```

ورود داده‌های پیش‌پردازش شده به شبکه عصبی کانولوشن

در مرحله بعدی پروژه، داده‌های پیش‌پردازش شده وارد مدل شبکه عصبی کانولوشنی خواهند شد که بخشی از معماری اصلی سیستم تشخیص نفوذ را تشکیل می‌دهد، این معماری سیستم تشخیص نفوذ ترکیبی که از لایه‌های کانولوشن، پولینگ و درخت تصمیم تشکیل شده است به صورت (شکل ۳-۳) است:



شکل ۱۰:لایه‌های معماری سیستم تشخیص نفوذ

نظر به اینکه شبکه عصبی کانولوشن بیشتر برای پردازش تصویر در ابعاد عرض، ارتفاع و عمق استفاده می‌شود و داده‌ها به صورت یک ماتریس دوبعدی هستند، و خروجی نمونه‌های مرحله پیش‌پردازش به صورت یک بردار یک‌بعدی هستند، قبل از ورود به شبکه عصبی، نمونه‌ها باید به شکل ماتریسی مناسب بازسازی شوند.

خروجی شکل داده‌ها بعد از تغییر شکل و قبل از ورود به شبکه عصبی کانولوشن به صورت زیر است:

ISSN: 2745-3673
http://www.Rcsj.ir

```
print(x_scale_reshaped.shape)
(125973, 40, 1)
```

نمونه‌های خروجی از مرحله پیش‌پردازش بر اساس مراحل سیستم تشخیص نفوذ وارد شبکه عصبی کانولوشنی شده که برای استخراج ویژگی‌ها، از لایه‌های کانولوشن و پولینگ و لایه فلاتن[1] عبور کرده است. برای ساخت این بخش از مدل، معماری را با استفاده کتابخانه کراس و وارد کردن مدل ترتیبی **sequential()**، لایه‌های کانولوشن و پولینگ وارد برنامه شده است.

برای همه لایه‌های کانولوشن تابع فعال‌سازی **"relu"** و لایه‌گذاری[2] از نوع **"same"** پیکربندی شده است که باعث می‌شود اطراف ماتریس به صورت متقارن سطر و ستون‌هایی با مقادیر صفر اضافه شود تا خروجی لایه کانولوشن تغییری نکند. اندازه گام را مشخص نکرده‌ایم، بنابراین پیش‌فرض آن در لایه‌های کانولوشن یک است.

اندازه ورودی را فقط برای لایه اول مشخص شده است، برای سایر لایه‌ها اندازه ورودی را از اندازه خروجی لایه قبل استنتاج می‌شود.

در لایه‌های پولینگ حداکثری پیش‌فرض گام ادغام، اندازه خود ادغام یعنی دو است. داده‌ها ابتدا وارد اولین لایه کانولوشن با تعداد فیلترهای لازم شده، به دنبال هر لایه کانولوشن یک‌لایه پولینگ آمده است که ابعاد ویژگی‌های به‌دست‌آمده از لایه کانولوشن را کاهش می‌دهد. تابع فعال‌سازی استفاده شده در این معماری تابع غیرخطی **ReLU** است که بلافاصله بعد از هر لایه کانولوشن می‌آید که خروجی‌های منفی را صفر می‌کند و بقیه را تغییر نمی‌دهد. این تابع نسبت به سایر توابع کارآمدتر است.اولین لایه کانولوشن در مدل شبکه عصبی با تعداد ۶۴ فیلتر و اندازه ۲ و تابع فعال سازی **Relu**، و لایه گذاری **padding** و شکل ورودی (۱و۴۰)، به صورت زیر است:

```
model=Sequential()
model.add(Conv1D(64,2, activation='relu', padding='same', input_shape=(40,1)))
```

دومین لایه، با پارامترهای مشابه لایه اول، بدون اندازه شکل ورودی به صورت زیر است:

```
model.add(Conv1D(64, 2, activation='relu', padding='same'))
```

سومین لایه، لایه پولینگ با اندازه ۲ بدون پارامتر به صورت زیر است: این لایه باعث می‌شود ابعاد خروجی لایه قبلی نصف شود.

```
model.add(MaxPooling1D(2))
```

لایه چهارم طبق معماری، لایه کانولوشن با اندازه ۲ و تعداد فیلتر ۶۴ برابر است با:

```
model.add(Conv1D(64, 2, activation='relu', padding='same'))
```

لایه بعدی کانولوشن با اندازه ۲ و تعداد فیلتر ۱۲۸ به صورت زیر است:

```
model.add(Conv1D(128, 2, activation='relu', padding='same'))
```

لایه بعدی لایه پولینگ با اندازه ۲ است.

```
model.add(MaxPooling1D(2))
```

در ادامه دولایه کانولوشن با ۱۲۸ فیلتر و پارامترهای کاملاً مشابه جهت یادگیری بیشتر ویژگی ها آمده است.

```
model.add(Conv1D(128,2, activation='relu', padding='same'))
model.add(Conv1D(128, 2, activation='relu', padding='same'))
```

ISSN: 2745-3673
http://www.Rcsj.ir

به دنبال این دولایه، لایه پولینگ از نوع مکس آمده است که ابعاد خروجی لایه کانولوشن را کاهش می‌دهد به این صورت که با لغزیدن روی ماتریس خروجی از لایه کانولوشن، مقادیر ماکزیمم را از محدوده لغزش انتخاب می‌کند و بقیه مقادیر را حذف می‌کند.

```
model.add(MaxPooling1D(2))
```

بعد از لایه‌های کانولوشن و پولینگ تابع حذف تصادفی[1] آمده است. این تابع به صورت تصادفی و با احتمال مشخصی برخی از ویژگی‌ها یا نورون‌ها را حذف یا غیرفعال می‌کند و باعث می‌شود که شبکه عمومی‌سازی بیشتری داشته باشد.

```
model.add(Dropout(0.5))
```

فرایند کانولوشن شش بار و پولینگ سه بار انجام می‌گیرد. در ادامه برای اینکه خروجی‌ها را صاف و به صورت یک بردار در بیاوریم از لایه فلاتن استفاده شده است.

می‌توان در حالت آموزش مدل، از یک تابع حذف تصادفی با یک احتمال مثلاً ۰/۵ بعد از لایه‌های کانولوشن و ادغام برای حذف برخی ویژگی ها استفاده کرد و مدل رو با تنظیماتی جهت آموزش پیکربندی کرد، یعنی از چه الگوریتمی برای بهینه‌سازی استفاده شود، از چه تابع زیانی استفاده کند، از چه معیارهای برای سنجش کارایی مدل استفاده شود. معمولا از تابع آدام[1] برای بهینه‌سازی، از تابع آنتروپی متقاطع دودویی[2] برای دسته بندی استفاده می شود. اما در این مرحله از پروژه چون هدف از لایه‌های کانولوشن و پولینگ فقط استخراج ویژگی‌هاست، پس نیازی به کامپایل و آموزش مدل در این مرحله نیست.

خروجی لایه‌های کانولوشن، ادغام همراه با پارامترهای تنظیم شده توسط **model.summar ()** به صورت زیر است:

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_12 (Conv1D)          (None, 40, 64)            192

 conv1d_13 (Conv1D)          (None, 40, 64)            8256

 max_pooling1d_6 (MaxPoolin  (None, 20, 64)            0
 g1D)

 conv1d_14 (Conv1D)          (None, 20, 64)            8256

 conv1d_15 (Conv1D)          (None, 20, 128)           16512

 max_pooling1d_7 (MaxPoolin  (None, 10, 128)           0
 g1D)

 conv1d_16 (Conv1D)          (None, 10, 128)           32896

 conv1d_17 (Conv1D)          (None, 10, 128)           32896

 max_pooling1d_8 (MaxPoolin  (None, 5, 128)            0
 g1D)

 dropout_2 (Dropout)         (None, 5, 128)            0

 flatten_2 (Flatten)         (None, 640)               0

=================================================================
Total params: 99008 (386.75 KB)
Trainable params: 99008 (386.75 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

در اینجا بخش اول پروژه به پایان می‌رسد.قبل از ورود به مدل دسته‌بندی درخت تصمیم، داده‌های استخراجی از مرحله اول، به مجموعه‌های آموزشی و آزمایشی تقسیم‌بندی می‌شوند. با استفاده کدهای زیر، مجموعه‌داده‌های آماده شده به نسبت ۸۰ به ۲۰ تقسیم بندی شده اند، ۸۰ درصد به آموزش مدل و ۲۰ درصد برای آزمایش تخصیص داده شده است.

```
x_train,x_test,y_train,y_test=train_test_split(features_x_scale,y,test_size=0.2)
```

ISSN: 2745-3673
http://www.Rcsj.ir

ورود ویژگی‌های استخراج شده به درخت تصمیم

برای ورود به بخش دوم معماری، ویژگی‌های استخراجی از بخش اول شبکه عصبی کانولوشن را وارد مدل درخت تصمیم می‌کنیم.ویژگی‌های استخراجی از لایه‌های کانولوشن و ادغام به عنوان ورودی به مدل طبقه‌بندی که درخت تصمیم هست تغذیه می‌شوند. چون ماتریس خروجی ویژگی‌ها به صورت سه‌بعدی هستند، و برای ورود به درخت تصمیم باید به صورت دوبعدی باشند، با استفاده از کدهای لازم به صورت یک ماتریس دوبعدی درآورده و برای ورود به بخش دوم پروژه که الگوریتم دسته‌بندی درخت تصمیم است آماده‌سازی شده است. ماتریس خروجی ویژگی‌های استخراجی بخش اول پروژه به صورت زیر است:

```
features_x_train
```

```
array([[9.95528768e-04, 4.94952966e-03, 9.76243347e-04, ...,
        1.35713257e-04, 0.00000000e+00, 1.23365957e-03],
       [0.00000000e+00, 1.03133470e-02, 0.00000000e+00, ...,
        0.00000000e+00, 3.26297777e-05, 2.77047948e-04],
       [0.00000000e+00, 1.01065282e-02, 0.00000000e+00, ...,
        0.00000000e+00, 0.00000000e+00, 3.38045240e-04],
       ...,
       [0.00000000e+00, 1.01065198e-02, 0.00000000e+00, ...,
        0.00000000e+00, 1.41643206e-04, 1.81154785e-04],
       [0.00000000e+00, 5.03295101e-03, 5.09499223e-04, ...,
        1.02131286e-04, 0.00000000e+00, 1.23365957e-03],
       [1.97011512e-03, 8.99971835e-03, 2.42279377e-04, ...,
        0.00000000e+00, 0.00000000e+00, 2.97593768e-04]], dtype=float32)
```

کدهای پایتون درخت تصمیم برای دسته‌بندی ویژگی‌های استخراجی از بخش اول به صورت زیر است:

در درخت تصمیم ابتدا روی ویژگی‌های استخراجی از بخش اول، آموزش می‌بینند، سپس مدل روی داده‌های آزمایشی پیش‌بینی شده است.

```
model_dt=DecisionTreeClassifier(criterion='entropy',splitter='best')
model_dt.fit(features_x_train,y_train)
y_pred=model_dt.predict(features_x_test)
```

در ادامه نتایج معیارهای دقت، صحت، فراخوانی و نمره $\mathbf{f_1}$ با مقایسه خروجی پیش‌بینی و خروجی اصلی و استفاده از کدهای زیر مشخص شده است.

```
y_pred=model_dt.predict(features_x_test)
accuracy=accuracy_score(y_test,y_pred)
print(accuracy)
recall=recall_score(y_test,y_pred)
print(recall)
f1=f1_score(y_test,y_pred)
print(f1)
precision=precision_score(y_test,y_pred)
print(precision)
```

در بخش دوم معماری پروژه تشخیص نفوذ از یک الگوریتم دسته‌بندی دوکلاسی استفاده شده است، لایه کاملاً متصل شبکه کانولوشن با طبقه‌بندی درخت تصمیم جهت عملکرد بهتر و دقت بیشتر جایگزین شده است. الگوریتم درخت تصمیم ویژگی‌های استخراجی را در دو کلاس حمله و نرمال طبقه‌بندی کرده است.

در این تحقیق از درخت تصمیم پایه **ID3** که از بهره اطلاعاتی و آنتروپی برای انتخاب ویژگی‌ها استفاده می‌کند، استفاده شده است.خروجی نهایی پروژه مدل تشخیص نفوذ همراه با تنظیمات پیکربندی در یکی از تجربه‌ها به صورت زیر است:

ISSN: 2745-3673
http://www.Rcsj.ir

نتایج معیارهای به ترتیب دقت، فراخوانی، نمره f_1 و صحت

با توجه به معماری تشخیص نفوذ، داده‌های مورداستفاده در این پژوهش نیاز به آماده‌سازی و پیش‌پردازش داشت، به همین جهت روی داده‌ها عملیات پیش‌پردازش از جمله نرمال‌سازی و مقیاس‌بندی انجام گرفت. سپس با فرمت مناسب جهت استخراج ویژگی‌ها وارد شبکه عصبی کانولوشن با تعداد تنظیم شده لایه‌های کانولوشن و پولینگ شدند، در ادامه ویژگی های خروجی شبکه عصبی کانولوشن به داده‌های آموزشی و آزمون تقسیم‌بندی، و در آخر جهت دسته‌بندی ویژگی‌ها به دو کلاس نفوذ و نرمال وارد درخت تصمیم شدند، و خروجی مدل هم دقت نسبتاً مناسبی داشت.

تشریح مجموعه‌داده

برای آموزش و سنجش سیستم‌های تشخیص نفوذ نیاز به مجموعه‌داده‌های مؤثری هست که بتواند کارایی و دقت سیستم را ارزیابی کند، دیتاست‌های مختلفی مانند **KDD cup** ، **DARPA** و **NSL-KDD** برای سیستم‌های تشخیص نفوذ وجود دارند، در این پروژه از مجموعه‌داده آموزشی **NSL-KDD** که نسخه ترمیم شده **KDD99**، و شامل ۱۲۵۹۷۳ رکورد است که از این تعداد ۶۷۳۴۳ مربوط به نمونه‌های عادی و نرمال است و بقیه یعنی ۵۸۶۳۰ رکورد مربوط به نفوذ یا حمله که از چهار نوع حمله به نام‌های **DOS**، **R2L**، **U2R** و **PRB** هستند، و برای مطالعه دقت و اثربخشی سامانه تشخیص نفوذ در تشخیص ناهنجاری‌ها[۲]در الگوهای ترافیک شبکه با استفاده از الگوریتم‌های ترکیبی یادگیری عمیق و درخت تصمیم استفاده شده است. این مجموعه داده شامل ترکیبی از ترافیک‌های نفوذ و نرمال است که شامل ۴۲ ویژگی است که ۴۱ ویژگی آن مربوط به اتصالات شبکه و یک ویژگی برچسب[۴] که مقادیر آن در پنج کلاس مختلف که چهار کلاس شامل رکوردهای حمله و یک کلاس عادی یا نرمال تعریف شده است. جزییات مجموعه داده استفاده شده در جدول زیرنشان داده شده است.

ISSN: 2745-3673
http://www.Rcsj.ir

جدول۳:دسته‌بندی انواع حملات در مجموعه‌داده

نوع حمله	تعداد
Normal	۶۷۳۴۳
DoS	۴۵۹۲۷
PRB	۱۱۶۵۶
R2L	۹۹۵
U2R	۵۲
Attacks	۵۸۶۳۰
Total	۱۲۵۹۷۳

حملات از نوع **Dos** که نفوذگر با استفاده از ارسال حجم زیاد درخواست‌ها، توان پاسخ‌گویی سرور را از کار می‌اندازد، شامل موارد زیر است:

Dos= ['back', 'land', 'neptune', 'pod', 'smurf', 'teardrop', 'apache2', 'mailbomb', 'processtable', 'udpstorm']

در حملات از نوع **Probe** ، نفوذگر با بررسی مداوم شبکه نقاط ضعف را پیدا کرده و از آن‌ها جهت حمله استفاده می‌کند و شامل موارد زیر است:

Probe= ['warezclient', 'warezmaster', 'spy', 'multihop', 'phf', 'ftp_write', 'guess_passwd', 'imap, xsnoop', 'xlock, worm', 'snmpguess','snmpgetattack', 'sendmail', 'named']

حملات از نوع **U2R** که نفوذگر ابتدا به عنوان یک کاربر نرمال قصد دسترسی و استفاده از شبکه دارد، پس از متصل‌شدن به شبکه، از نقاط ضعف برای دسترسی به سیستم آسیب‌پذیر استفاده می کند و به عنوان ریشه کنترل آن را به دست می‌گیرد، شامل حملات زیر است:

U2R= ['portsweep', 'satan', 'ipsweep', 'nmap', 'mscan', 'saint']

و حملات از نوع **R2L** در این حمله نفوذگر توانایی ارسال بسته به یک سیستم در شبکه دارد اما روی سیستم حساب کاربری ندارد از نقاط ضعف سامانه برای دسترسی به عنوان کاربر محلی به سیستم قربانی استفاده می‌کند و شامل موارد زیر است:

R2L= ['rootkit', 'perl', 'loadmodule', 'buffer_overflow', 'httptunnel', 'ps', 'sqlattack', 'xterm']

در کلاس‌بندی دادها برای تشخیص نفوذ معمولاً دو شیوه متفاوت وجود دارد:

دسته‌بندی دوکلاسی که داده‌ها در دو کلاس نرمال و نفوذ تقسیم‌بندی می‌شود و دسته‌بندی پنج کلاسی.در این مقاله که رسالت آن افزایش دقت تشخیص نفوذ است، از دسته‌بندی دودویی، در دو کلاس نرمال و نفوذ استفاده شده است.

معیارهای ارزیابی مدل

برای ارزیابی کارایی مدل‌های طبقه‌بندی از جمله سیستم‌های تشخیص نفوذ، معیارهای مانند دقت، صحت، فراخوانی و **F1** همراه با ترکیب‌های مختلفی از کلاس واقعی و کلاس‌های پیش‌بینی مورد بررسی قرار گرفته است.ابتدا مفاهیم مقادیر ماتریس درهم‌ریختگی که از ترکیب مقادیر معیارها به دست می‌آیند را بیان می‌کنیم.

- مثبت راستین[1](**TP**): یعنی پیش‌بینی درست نفوذ، مدل داده‌های نفوذ را درست پیش‌بینی کرده و در کلاس واقعی خودش دسته‌بندی کرده است. به تعریف دیگر تعداد نمونه‌های نفوذ که درست پیش‌بینی شده است.

ISSN: 2745-3673
http://www.Rcsj.ir

- منفی راستین[۲](**TN**): پیش‌بینی درست داده‌های عادی یا نرمال، یعنی داده‌های نرمال، در کلاس درست دسته‌بندی شده‌اند. بعبارتی تعداد نمونه‌های نرمال که به درستی توسط مدل تشخیص داده شده است.
- مثبت کاذب[۳](**FP**): یعنی پیش‌بینی نادرست داده‌های نرمال. داده‌های نرمال به اشتباه در کلاس نفوذ طبقه‌بندی شده است. یعنی تعداد رکوردهای عادی که به صورت نفوذ تشخیص داده است.
- منفی کاذب[۴](**FN**): یعنی پیش‌بینی نادرست داده‌های حمله. داده‌های حمله در کلاس نرمال دسته‌بندی شده است. تعداد رکوردهای نفوذ که به اشتباه به صورت نرمال تشخیص داده است.

معیار دقت: رایج‌ترین معیار است و بیانگر آن است مدل تشخیص نفوذ، چند درصد از نمونه‌های آزمون در مجموعه‌داده‌ها را درست تشخیص داده است به عبارتی چه تعداد از نمونه‌های جدید نفوذ و نرمال را درست پیش‌بینی کرده است که برابر با نسبت مجموع تعداد پیش‌بینی‌های درست حمله و نرمال به کل نمونه‌های ورودی است. رابطه آن به صورت زیر است:

رابطه معیار دقت

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

اگر چه مهم‌ترین معیار در یک مدل، نرخ دقت است، اما کافی نیست، و به تنهایی نمی‌تواند معیار معتبری برای کارایی یک مدل باشد. تمرکز ما عمدتاً روی این است که مدل چقدر قابل‌اعتماد است، یعنی مدل چقدر روی داده‌ها جدید انعطاف‌پذیری و قابلیت تعمیم دارد، برای همین به معیارهای دیگری نیز نیاز است.

معیار صحت: چون معیار دقت یک نقطه‌ی ضعفی دارد و نمی‌تواند تفاوتی بین خطاهای منفی کاذب و مثبت کاذب قائل شود و آنها را یکسان می‌گیرد، جهت غلبه بر این مشکل از معیار صحت استفاده می‌شود که برابر است با درصدی از موارد مثبت راستین در بین نمونه‌های که به عنوان مثبت پیش‌بینی‌شده است، به عبارتی این معیار می‌گوید چند درصد یا چه تعداد از تشخیص نفوذها درست بوده است؛ چون ممکن است تعدادی از تشخیص نفوذها اشتباه بوده باشند. از نظر رابطه ریاضی برابر است با نسبت مثبت واقعی به مجموع مثبت واقعی و مثبت کاذب است

معیار فراخوانی: از معیارهای مهم دیگر تشخیص نفوذ است که بیشتر روی نفوذ تأکید دارد و برابر است با نسبت تعداد حملات شناسایی شده توسط سیستم به تعداد کل حملات است. یعنی چند درصد از نفوذها را درست تشخیص داده، امکان دارد بعضی از نفوذها را به اشتباه، به صورت نرمال تشخیص داده باشد. از نظر فرمول ریاضی نسبت مثبت واقعی به مجموع مثبت واقعی و منفی کاذب است.

$$\text{Precision} = \frac{TP}{TP + FP}$$

رابطه معیار فراخوانی

$$\text{Recall} = \frac{TP}{TP+FN}$$

معیار **F1**: در بعضی موارد معیارهای صحت و فراخوانی به تنهایی برای مدل کارایی ندارند، برای همین از این معیار برای تعادل بین معیارهای صحت و فراخوانی استفاده می‌شود و نشان می‌دهد طبقه‌بندی چقدر دقیق است، هر چه مقدار **F1** بیشتر باشد کارایی مدل بهتر است.

رابطه معیار **F1**

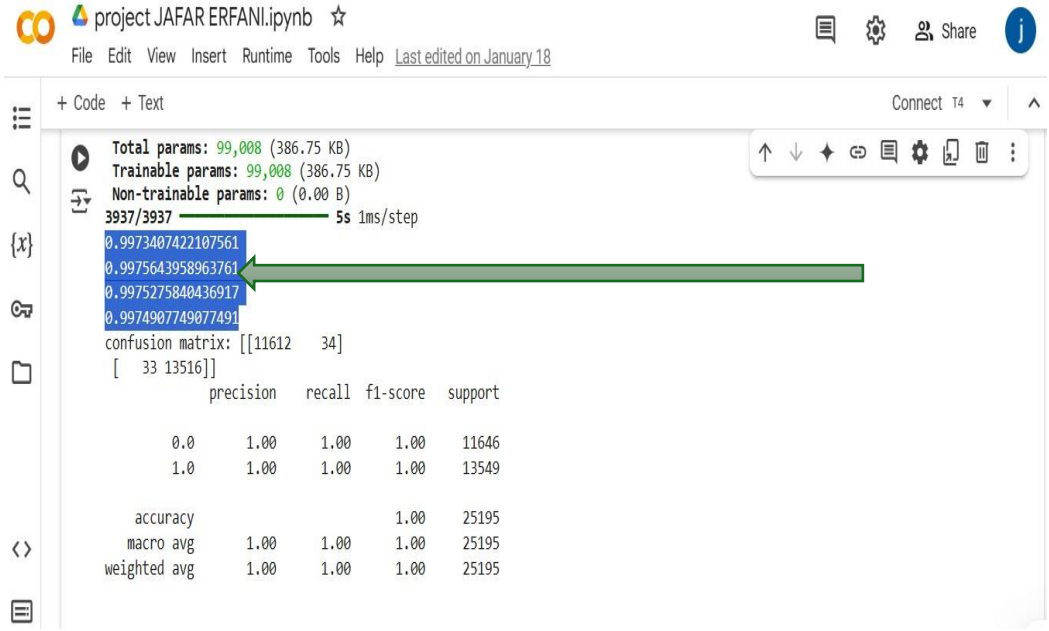
$$f1\ score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right)$$

تحلیل و ارزیابی مدل پیشنهادی

پیاده‌سازی این پروژه با زبان پایتون[۱] و کتابخانه‌های معروف یادگیری عمیق مانند فریم‌ورک تنسور فلو[۲]، کراس[۳]، و سایکت لرن[۴]، در محیط ژوپیتر نوت‌بوک[۵] گوگل کولاب[۶]، با پردازشگر **T4 Gpu** و رم ۱۲ گیگ انجام شده است. مجموعه‌داده برای آموزش و آزمایش به نسبت ۸۰ به ۲۰ تقسیم شده، و تابع فعال‌سازی غیرخطی **ReLU** در تمامی لایه‌های کانولوشن برای

ISSN: 2745-3673
http://www.Rcsj.ir

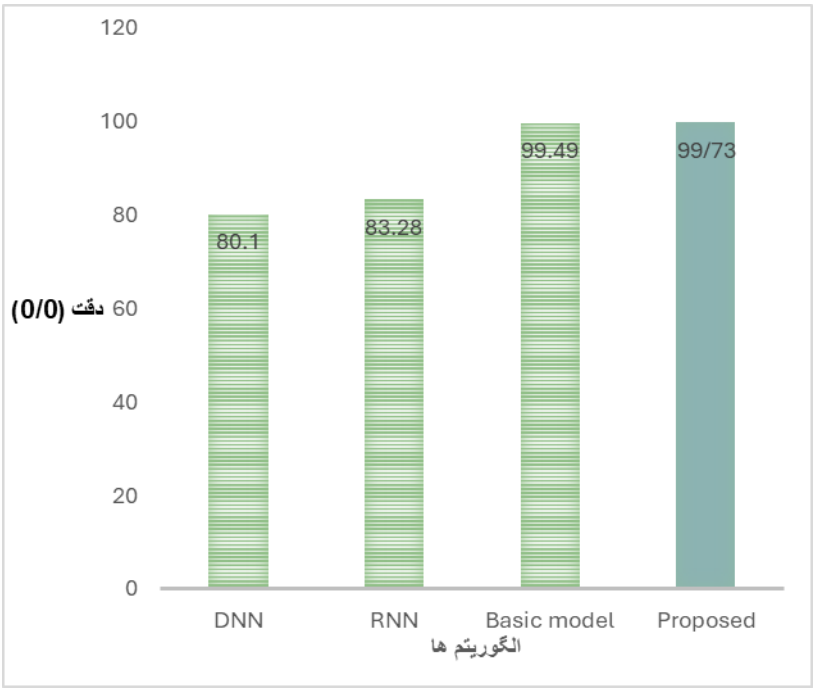
فرایند استخراج ویژگی‌ها استفاده شده است. ابتدا داده‌ها وارد شبکه عصبی کانولوشن شده، با استفاده از لایه‌های کانولوشن و ادغام، ویژگی‌های مهم و بهینه استخراج شده است، ویژگی‌های خروجی از شبکه عصبی کانولوشن توسط لایه مسطح فلاتن به صورت یک بردار درآمدند، سپس برای دسته‌بندی وارد درخت تصمیم شده اند. در درخت تصمیم با استفاده از داده‌های آموزشی و تابع فیت[۷] مدل آموزش دیده است، سپس روی داده‌های دیده نشده آزمایشی با استفاده از تابع **predict** پیش‌بینی شده است. در آخر با استفاده مقادیر ماتریس درهم‌ریختگی، عملکرد مدل پیشنهادی به وسیله معیارهایی مانند دقت، صحت، بازخوانی و $\mathbf{F_1}$ مورد ارزیابی قرار گرفته که به ترتیب، ۹۹/۷۳، ۹۹/۷۵، ۹۹/۷۵، ۹۹/۷۴ تجربه شده است.



نتایج معیارهای به ترتیب، دقت، فراخوانی، نمره F1و صحت

برای ارزیابی عملکرد مدل پیشنهادی نسبت به سایر، از مدل‌های شبکه عصبی عمیق، شبکه عصبی بازگشتی، و مدل پایه که از مجموعه معیارهای یکسانی استفاده می‌کنند در تحلیل مقایسه‌ای استفاده شده است.تحلیل مدل پیشنهادی و سه مدل دیگر از نظر دقت را نشان می‌دهد. دقت تشخیص مدل پیشنهادی بر اساس دسته بندی درست داده ها نسبت به تعداد کل نمونه ها، اندازه گیری می‌شود. یعنی این مدل پیشنهادی توانسته است ۰/۹۹۷۳ نمونه ها را درست تشخیص دهد و ۰/۲۷ درصد در این معیار خطا داشته است با این دقت به دست آمده، تقریبا نزدیک به صد درصد نمونه ها در کلاس درست دسته بندی کرده است. این دقت نسبت به مدل **DNN** حدوداً ۱۹ درصد و نسبت به مدل **RNN** حدوداً ۱۶ درصد و نسبت به مدل پایه حدوداً ۲۴ صدم درصد بیشتر شده است.

ISSN: 2745-3673
http://www.Rcsj.ir



نمودار۱:مقایسه معیار دقت بین الگوریتم‌ها

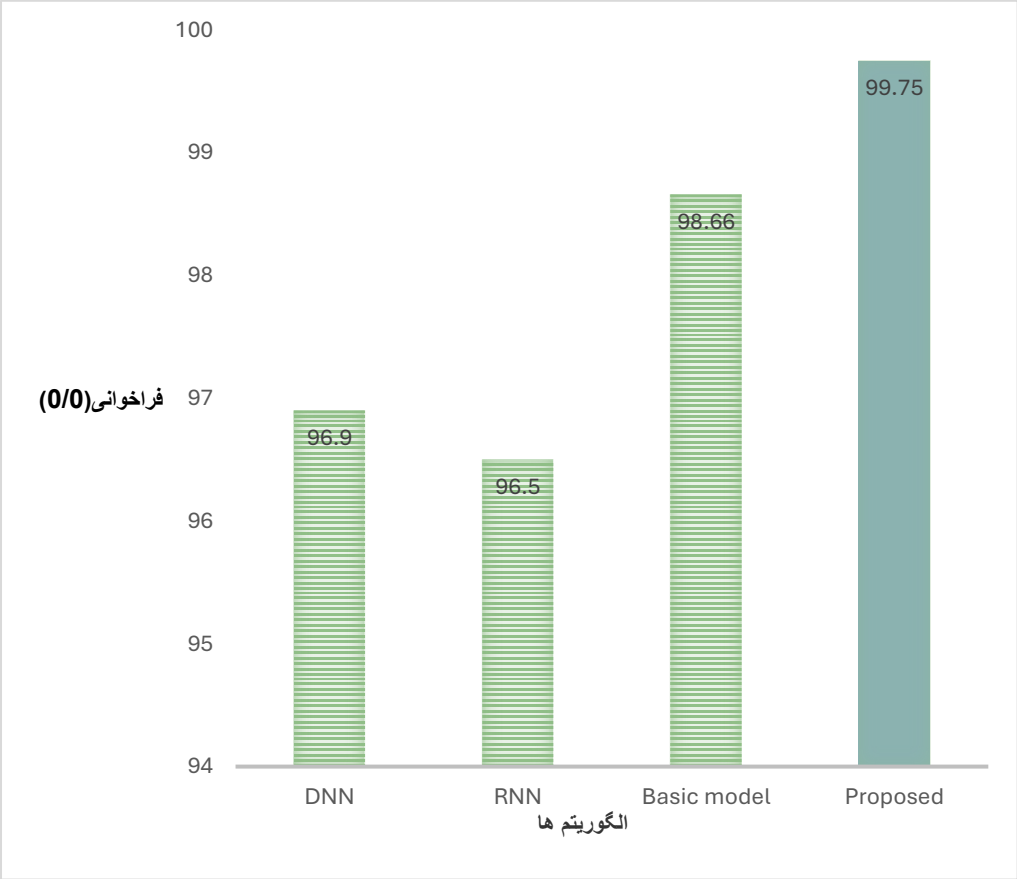
تحلیل ارزیابی مدل پیشنهادی از نظر معیار صحت با سه مدل دیگر نشان می‌دهد.معیار صحت به دست آمده حدوداً ۰/۹۹۷۵ است به عبارتی این تعداد یا درصد از نفوذ را درست تشخیص داده است و درصد کمی از تشخیص نفوذها اشتباه بوده است که نسبت به مدل **DNN** تفاوت قابل‌توجهی دارد، و نسبت به مدل **RNN** تفاوت اندکی در حدود سه درصد دارد و نسبت به مدل پایه حدوداً یک درصد بیشتر است. یعنی تقریباً همه تشخیص نفوذها توسط مدل پیشنهادی درست بوده است و درصد کمی خطای تشخیص دارد.

نمودار۲:مقایسه معیار صحت بین الگوریتم‌ها

ارزیابی مدل از نظر معیار فراخوانی نسبت به سه مدل دیگر نشان می‌دهد. از تحلیل بدست آمده می‌توان نتیجه گرفت که معیار فراخوانی مدل پیشنهادی نسبت به مدل **DNN** حدوداً ۳ درصد، و نسبت به مدل **RNN** حدوداً ۳/۵ درصد، و نسبت به

ISSN: 2745-3673
http://www.Rcsj.ir

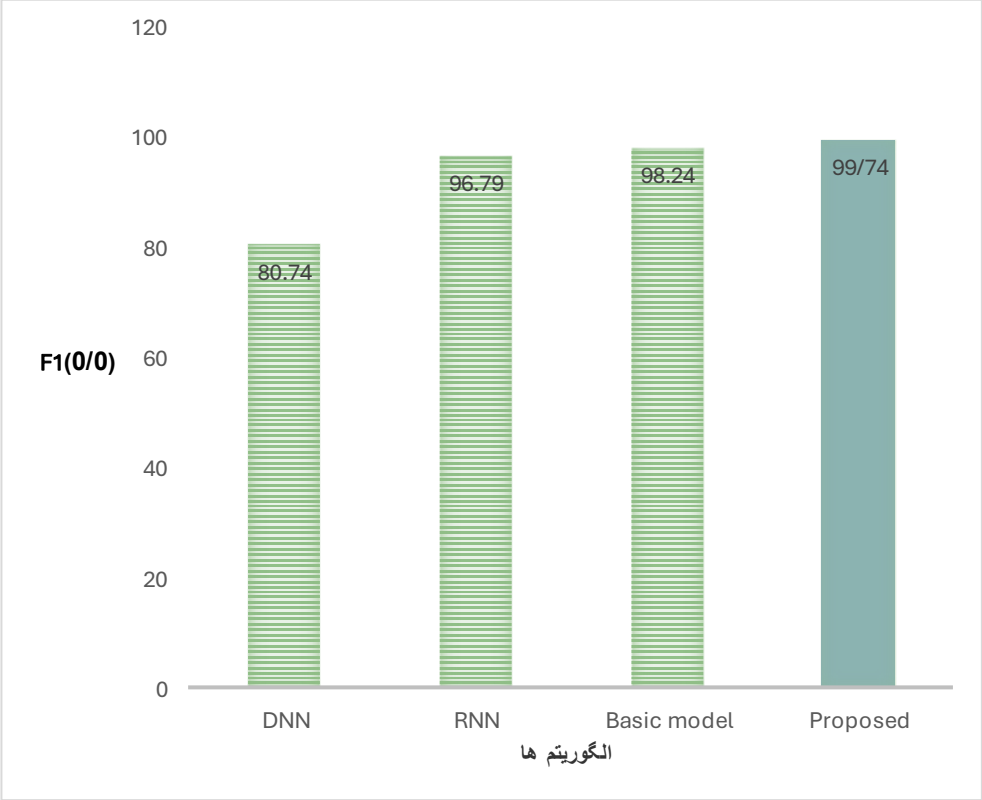
مدل پایه حدود ۱ درصد بیشتر است و این یعنی مدل پیشنهادی توانسته است درصد بالایی از نفوذ های را درست تشخیص دهد و درصد کمی از نفوذها را اشتباه تشخیص داده و این یعنی اطمینان و اعتماد به مدل تشخیص را بیشتر می کند.



نمودار۳:مقایسه معیار فراخوانی بین الگوریتم‌ها

بر اساس معیارهای صحت و فراخوانی، معیار **F1** محاسبه می‌شود. این معیار در سیستم پیشنهادی نسبت به مدل **DNN** حدوداً ۱۹ درصد و حدوداً ۳ درصد نسبت به مدل **RNN** و حدود یک درصد نسبت به مدل پایه بیشتر است. و این معیار نشان می دهد کارایی مدل نسبت به سایر مدل های مورد مقایسه بیشتر است و این میزان دقیق بودن مدل طبقه بندی را می رساند.

ISSN: 2745-3673
http://www.Rcsj.ir



نمودار۴:مقایسه معیار **F1** بین الگوریتم‌ها

نرخ خطای[1] مدل پیشنهادی بر اساس رابطه خطا برابر با ۰/۰۰۲ است، درحالی‌که نرخ خطا برای مدل **DNN**، ۰/۱۹ و برای مدل **RNN**، ۰/۱۷ و نرخ خطای مدل پایه ۰/۰۰۵ محاسبه می‌شود، پس نرخ خطا مدل پیشنهادی نسبت مدل‌های دیگر کمتر است.

رابطه نرخ خطا (۴-۵)

$$Error\ rate = 1 - Accuracy$$

نتایج تجزیه و تحلیل تجربی در مورد معیارهای ارزیابی مدل نسبت به مدل‌های دیگر به صورت خلاصه گردآوری شده است. با توجه به داده های جدول، مشاهده می‌شود که استفاده از مدل ترکیبی شبکه عصبی کانولوشن و درخت تصمیم در مقایسه با مدل‌های غیرترکیبی شبکه‌های عصبی مانند شبکه عصبی عمیق و شبکه عصبی بازگشتی، به خاطر استخراج بهینه ویژگی‌ها، دقت بیشتری در تشخیص نفوذ نسبت به مدل‌های ساده هم تراز خود دارد. با نظر به عملکرد بدست آمده از مدل پیشنهادی در فضای آزمایشگاهی، این مدل برای سیستم‌های تشخیص نفوذ در شبکه‌های اینترنت اشیا همراه با ملاحضاتی می‌تواند مناسب باشد، اگرچه پویایی در شبکه اینترنت اشیا باعث می شود سیستم پیشنهادی در حالت واقعی خطای نسبتاً بیشتری نسبت به مدل تحقیقاتی داشته باشد.

جدول ۴: تحلیل معیارهای عملکرد مدل پیشنهادی به درصد

الگوریتم‌ها	دقت	صحت	فراخوانی	نمره **F1**
DNN [33]	۸۰/۱	۶۹/۲۰	۹۰/۹۶	۸۰/۷۴
RNN [34]	۸۳/۲۸	۹۶/۵۰	۹۷/۰۹	۹۶/۷۹
CNN-DT [29]	۹۹/۴۹	۹۸/۶۶	۹۸/۲۴	۹۸/۴۴
Proposed	۹۹/۷۳	۹۹/۷۵	۹۹/۷۵	۹۹/۷۴

ISSN: 2745-3673
http://www.Rcsj.ir

بحث و نتیجه‌گیری

برای شبکه گسترده ای مانند اینترنت اشیا که سهم عمده‌ای در هوشمندی دنیای آینده بشریت دارد، مهم‌ترین مسئله، امنیت این شبکه است. تحقیقات گسترده‌ای در زمینه امنیت شبکه‌ها از جمله، شبکه اینترنت اشیا، با رویکردهای متفاوت در بعد آزمایشگاهی انجام شده است، در این پایان‌نامه یک سیستم تشخیص نفوذ ترکیبی با استفاده از ترکیب شبکه عصبی کانولوشنی برای استخراج ویژگی داده‌ها و درخت تصمیم برای دسته‌بندی دودویی ویژگی‌های استخراجی، جهت تعیین اینکه کدام یک از نمونه ها در کلاس نرمال هستند و کدام یک در کلاس نفوذ یا حمله هستند، انجام شده است.این مقاله با داده‌های **NSL-KDD** در بستر گوگل کولاب انجام شده، و جهت ارزیابی نتایج از معیارهای دقت، صحت، یادآوری و **F1** استفاده شده است که به ترتیب تا مقادیر ۹۹/۷۳، ۹۹/۷۵، ۹۹/۷۵، ۹۹/۷۴ درصد تجربه شده است.

در این مقاله که هدف آن افزایش دقت سیستم تشخیص نفوذ در شبکه اینترنت اشیا بوده، از دیتاست **NSL-KDD** که مهم‌ترین مجموعه داده نفوذ محسوب می‌شود استفاده شده است، این دادها نیاز به پیش‌پردازش و نرمال‌سازی داشتند و جهت افزایش دقت مدل تشخیص نفوذ، بر خلاف روش‌های معمول که داده‌ها در چهار کلاس حمله و یک کلاس نرمال تقسیم‌بندی می‌شدند، برچسب یا خروجی داده‌ها در دو کلاس نفوذ و نرمال دسته‌بندی شده است. سپس داده‌ها را وارد شبکه عصبی کانولوشن با تعداد لایه‌های مناسب و تنظیم دستی فراپارامترها کرده، ویژگی‌های استخراجی در این مرحله، وارد مدل تصمیم‌گیری دوگانه درخت تصمیم شدند، معیارهای ارزیابی به‌دست‌آمده با سه تحقیق پیشین و هم معیار مقایسه شده است، از این مقایسه می‌توان نتیجه گرفت که نسبت به سیستم های تشخیص نفوذ تک مدلی شبکه‌های عصبی دقت بیشتری دارد.

با توجه به نتایج به دست آمده از ارزیابی معیارها و مقایسه با روش‌های معمول، سه نتیجه کلی دارد: اول اینکه استفاده از سیستم‌های تشخیص نفوذ ترکیبی عملکرد بهتری دارند. دوم می‌توان از یک مدل دسته‌بندی مانند درخت تصمیم به جای لایه کاملاً متصل شبکه عصبی کانولوشن برای کلاس‌بندی ویژگی‌ها استفاده کرد و کارایی مدل را افزایش داد و آخر اینکه استفاده از الگوریتم دسته‌بندی دودویی به جای دسته‌بندی چند کلاسی در این سیستم‌ها می‌تواند دقت تشخیص نفوذ را افزایش دهد.

پیشنهادهای تحقیقات آینده

اگر چه پژوهش‌های زیادی در زمینه سیستم‌های تشخیص نفوذ که وظیفه اصلی آنها، تشخیص نفوذ و گزارش آن به مدیران است، انجام شده، اما هنوز این پژوهش ها در مراحل تحقیقاتی و آزمایشی هستند ولی می‌توان از نتایج این تحقیق در صورت وجود زیرساخت‌های نرم‌افزاری و سخت افزاری لازم، در جاهایی که اینترنت اشیا در بستر هوش مصنوعی استفاده می‌شود، به کار گرفت. برای تحقیقات آینده، با توجه به این که سیستم‌های تشخیص نفوذ، فقط توانایی تشخیص نفوذ دارند، و معمولاً قادر به مقابله با نفوذ نیستند، پس کارآمدتر و به‌صرفه است سیستم‌هایی مورد تحقیق و پژوهش قرار گیرند که ترکیبی از سیستم‌های تشخیص نفوذ و سیستم‌های مقابله با نفوذ در یک سیستم واحد باشد تا باعث کاهش هزینه سیستم های امنیتی و هم کاهش فرآیند شناسایی و مقابله با نفوذ و هم نیاز به زیرساخت‌های مجزا نخواهد بود و همین‌طور تحقیق در باره‌ی سیستم‌های هوشمند ردیابی نفوذ جهت کشف منبع نفوذ که می‌تواند در صورت امکان منابع نفوذی را از کار بیندازد و ضریب امنیتی را بیشتر کند می‌تواند موضوع تحقیقات آینده پژوهشگران باشد.

ISSN: 2745-3673
http://www.Rcsj.ir

منابع

[۱] باباگلی، م. (۱۴۰۲). ارائه یک الگوی فراابتکاری تشخیص نفوذ به کمک انتخاب ویژگی مبتنی بر بهینه سازی گرگ خاکستری بهبود یافته و جنگل تصادفی. سینگال و پردازش داده ها، دوره ۲۰ ، شماره ۱ ، ص ۱۴۴-۱۳۳.

[۲] تجری، س الف. (۱۴۰۰). تشخیص نفوذ در شبکه های رایانه ای با استفاده از درخت تصمیم و کاهش ویژگی ها. نشریه علمی پدافند الکترونیکی و سایبری، دوره ۹ ، شماره ۳ ، ص ۱۰۸-۹۹.

[۳] زنده دل، م. حمیدزاده، ج. (۱۴۰۲). بهبود تشخیص نفوذ به شبکه اینترنت اشیا با استفاده از یادگیری عمیق و الگوریتم بهینه سازی میگوی آشوبی. مجله مهندسی برق دانشگاه تبریز، دوره ۵۳ ، شماره ۲ ، ص ۱۳۸-۱۲۷.

[۴] طاهری، ر.، پارسائی، م ر.، و جاویدان، ر. (۱۳۹۶). سامانه تشخیص نفوذ بلادرنگ با استفاده از ترکیب گسسته سازی و انتخاب ویژگی های مهم. مجله علمی پژوهشی علوم و فناوری های پدافند نوین، دوره ۸ ، شماره ۳ ، ص ۲۶۳-۲۵۱.

[۵] مظلوم، ج. بیگدلی، ح. (۱۴۰۱). شبکه عصبی عمیق ترکیبی بهینه ادغام شده با انتخاب ویژگی برای سامانه تشخیص نفوذ در حملات سایبری. نشریه علمی پدافند الکترونیکی و سایبری، دوره ۱۰ ، شماره ۴ ، ص ۵۱-۴۱.

[۶] میلاد، ن. (۱۳۹۹). یادگیری عمیق: اصول، مفاهیم و رویکردها (چاپ اول). تهران: نشر میعاد اندیشه ص ۲۵۴.

[۷] محمدزاده، الف. (۱۳۹۸). تکمیل تصویر با استفاده از یادگیری عمیق، پایان نامه کارشناسی ارشد، گروه کامپیوتر، دانشکده مهندسی، دانشگاه شهید چمران اهواز.

[8] Abd Elaziz, M., Al-qaness, M. A., Dahou, A., Ibrahim, R. A., Abd El-Latif, A. A. (2023). Intrusion detection approach for cloud and IoT environments using deep learning and Capuchin Search Algorithm. *Advances in Engineering Software*, *176*,103402

[9] Aldweesh, A., Derhab, A., Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems:A survey,taxonomy,and open issues. *Knowledge-Based Systems*, *189*, 105124.

[10] Burhan, M., Rehman, R. A., Khan, B., Kim, B. S. (2018). IoT elements, layered architectures and security issues: A comprehensive survey. *sensors*, *18*(9), 2796.

[11] C Müller, A. (2017). Introduction to machine learning with python.

[12] De Souza, C. A., Westphall, C. B., Machado, R. B. (2022). Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments. *Computers & Electrical Engineering*, *98*, 107694

[13] Denning, D. E. (2015). Toward more secure software. *Communications of the ACM*, *58*(4), 24-26.

[14] Islam, M. Z., Islam, M. M., Asraf, A. (2020). A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Informatics in medicine unlocked*, *20*, 100412.

[15] Imran, S., Khan, M. U. G., Idrees, M., Muneer, I., Iqbal, M. M. (2018). An enhanced framework for extrinsic plagiarism avoidance for research article. *Technical Journal*, *23*(01), 84-92.

[16] Jyothsna, V. V. R. P. V., Prasad, R., Prasad, K. M. (2011).
A review of anomaly based intrusion detection systems. *International Journal of Computer Applications*, *28*(7), 26-35.

[17] Kabiri, P., Ghorbani, A. A. (2005). Research on intrusion detection and response: A survey. *Int. J. Netw. Secur.*, *1*(2), 84-102.

[18] Kumar, D. A., Venugopalan, S. (2017). Intrusion detection systems: a review. *International Journal of Advanced Research in Computer Science*, *8*(8), 356-370

[19] Labonno, M., & Ahmed, S. (2023). *Network Intrusion Detection Using Machine Learning & Deep Learning* (Doctoral dissertation, East West University).

[20] Malathi, C., Padmaja, I. N. (2023). Identification of cyber attacks using machine learning in smart IoT networks. *Materials Today: Proceedings*, *80*, 2518-2523.

ISSN: 2745-3673
http://www.Rcsj.ir

[21] Mishra, P., Varadharajan, V., Tupakula, U., Pilli, E. S. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE communications surveys & tutorials*, *21*(1), 686-728

[22] Pingale, S. V., Sutar, S. R. (2022). Remora whale optimization-based hybrid deep learning for network intrusion detection using CNN features. *Expert Systems with Applications*, *210*, 118476.

[23] Potluri, S., Ahmed, S., Diedrich, C. (2018). Convolutional neural networks for multi-class intrusion detection system. In *Mining Intelligence and Knowledge Exploration: 6th International Conference, MIKE 2018, Cluj-Napoca, Romania, December 20–22, 2018, Proceedings 6* (pp. 225-238). Springer International Publishing

[24] Simon, J., Kapileswar, N., Polasi, P. K., Elaveini, M. A. (2022). Hybrid intrusion detection system for wireless IoT networks using deep learning algorithm. *Computers and Electrical Engineering*, *102*, 108190.

[25] Satam, P., Hariri, S. (2020). WIDS: An anomaly based intrusion detection system for Wi-Fi (IEEE 802.11) protocol. *IEEE Transactions on Network and Service Management*, *18*(1), 1077-1091

[26] Srinivasan, C. R., Rajesh, B., Saikalyan, P., Premsagar, K., Yadav, E. S. (2019). A review on the different types of Internet of Things (IoT). *Journal of Advanced Research in Dynamical and Control Systems*, *11*(1), 154-158.

[27] Tang, M., Alazab, M., Luo, Y., Donlon, M. (2018). Disclosure of cyber security vulnerabilities: time series modelling. *International Journal of Electronic Security and Digital Forensics*, *10*(3), 255-275.

[28] Tsai, C. F., Hsu, Y. F., Lin, C. Y., Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *expert systems with applications*, *36*(10), 11994-12000

[29] Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. IEEE
Accss 2019;7:41525–50.

[30] Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 2017;5:21954–61.

[31] Zarpelão, B. B., Miani, R. S., Kawakani, C. T., De Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, *84*, 25-37.

[32] Annual number of malware attacks worldwide from 2015 to 2023 https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide/

[33] Convolutional Neural Network. (2019). Retrieved from https://towardsdatascience.com/ covolutional-neural-network cb0883dd6529