دو فصلنامه علمی
فناوری اطلاعات و ارتباطات ایران

سال پانزدهم، شماره ۵۷ و۵۸ ، پاییز و زمستان ۱۴۰۲
صفحات:۱۶۵ الی ۱۹۰

Improvement of intrusion detection system on Industrial Internet of Things based on deep learning using metaheuristic algorithms

Mohammadreza Zeraatkar Moghaddam[*], Majid Ghayoori[**]
[*] M.Sc., Imam Hossein University, Tehran, Iran
[**] Assistant Professor, Imam Hossein University, Tehran, Iran

Abstract
Due to the increasing use of industrial Internet of Things (IIoT) systems, one of the most widely used security mechanisms is intrusion detection system (IDS) in the IIoT. In these systems, deep learning techniques are increasingly used to detect attacks, anomalies or intrusions. In deep learning, the most important challenge for training neural networks is determining the hyperparameters in these networks. To overcome this challenge, we have presented a hybrid approach to automate hyperparameter tuning in deep learning architecture by eliminating the human factor. In this article, an IDS in IIoT based on convolutional neural networks (CNN) and recurrent neural network based on short-term memory (LSTM) using metaheuristic algorithms of particle swarm optimization (PSO) and Whale (WOA) is used. This system uses a hybrid method based on neural networks and metaheuristic algorithms to improve neural network performance and increase detection rate and reduce neural network training time. In our method, considering the PSO-WOA algorithm, the hyperparameters of the neural network are determined automatically without the intervention of human agent. In this paper, UNSW-NB15 dataset is used for training and testing. In this research, the PSO-WOA algorithm has use optimized the hyperparameters of the neural network by limiting the search space, and the CNN-LSTM neural network has been trained with this the determined hyperparameters. The results of the implementation indicate that in addition to automating the determination of hyperparameters of the neural network, the detection rate of are method improve 98.5, which is a good improvement compared to other methods.

Keywords: intrusion detection system, Industrial Internet of Things, metaheuristic algorithms, neural networks

بهبود سیستم تشخیص نفوذ در اینترنت‌اشیاء صنعتیِ مبتنی بر یادگیری عمیق با استفاده از الگوریتم‌های فراابتکاری

محمدرضا زراعت‌کار مقدم*ˣ، مجید غیوری**

*کارشناس ارشد، دانشگاه جامع امام حسین(ع)

**استادیار دانشکده فناوری اطلاعات و ارتباطات دانشگاه جامع امام حسین(ع)

تاریخ دریافت:۱۴۰۱/۰۸/۱۵ تاریخ پذیرش:۱۴۰۱/۱۱/۰۹

نوع مقاله: پژوهشی

چکیده

با توجه به گسترش روز افزون استفاده از سامانه‌های اینترنت‌اشیاء صنعتی یکی از پرکابردترین مکانیزم‌های امنیتی، سیستم‌های تشخیص نفوذ در اینترنت‌اشیاء صنعتی می‌باشد. در این سیستم‌ها از تکنیک‌های یادگیری عمیق به‌طور فزآینده‌ای برای شناسایی حملات، ناهنجاری‌ها یا نفوذ استفاده می‌شود. در یادگیری عمیق مهم‌ترین چالش برای آموزش شبکه‌های عصبی، تعیین فراپارامترهای اولیه در این شبکه‌ها است. ما برای غلبه بر این چالش، به ارائه‌ی رویکردی ترکیبی برای خودکارسازی تنظیم فراپارامتر در معماری یادگیری عمیق با حذف عامل انسانی پرداخته‌ایم. در این مقاله یک سیستم تشخیص نفوذ در اینترنت‌اشیاء صنعتی مبتنی بر شبکه‌های عصبی کانولوشن (CNN) و شبکه عصبی بازگشتی مبتنی بر حافظه کوتاه مدت (LSTM) با استفاده از الگوریتم‌های فراابتکاری بهینه‌سازی ازدحام ذرات (PSO) و وال (WOA) ارائه شده است. این سیستم یک روش ترکیبی براساس شبکه‌های عصبی و الگوریتم‌های فراابتکاری برای بهبود عملکرد شبکه عصبی در راستای افزایش نرخ تشخیص و کاهش زمان آموزش شبکه‌های عصبی می‌باشد. در روش ما با درنظر گرفتن الگوریتم PSO-WOA، فراپارامترهای شبکه عصبی بدون دخالت عامل انسانی و به‌صورت خودکار تعیین شده است. در این مقاله از مجموعه‌داده‌ی UNSW-NB15 برای آموزش و آزمایش استفاده شده است. در این پژوهش، الگوریتم PSO-WOA با محدود کردن فضای جستجو، فراپارامترهای شبکه عصبی را بهینه کرده و شبکه عصبی CNN-LSTM با فراپارامترهای تعیین شده آموزش دیده است. نتایج پیاده‌سازی حکایت از آن دارد که علاوه بر خودکارسازیِ تعیین فراپارامترهای شبکه‌ی عصبی، نرخ تشخیص روش ما ۹۸,۵ درصد بوده که در مقایسه با روش‌های دیگر بهبود مناسبی داشته است.

واژگان کلیدی: سیستم تشخیص نفوذ، اینترنت‌اشیاء صنعتی، شبکه‌های عصبی، الگوریتم‌های فراابتکاری

ˣ نویسنده مسئول: محمدرضا زراعت‌کار مقدم mzeraatkar@ihu.ac.ir

۱. مقدمه

اینترنت‌اشیاء صنعتی[1] (IIoT) همان بهره‌گیری از فناوری اینترنت‌اشیاء در سیستم‌های کنترل صنعتی[2] (ICS) است. سیستم‌های کنترل صنعتی بخشی جدایی ناپذیر از زیرساخت‌های مهم هستند و مدت زمان طولانی است که برای نظارت بر ماشین آلات و فرآیندهای صنعتی مورد استفاده قرار گرفته‌اند. برای تجهیزات و فرآیندهای صنعتی که نیاز به کنترل دارند، شبکه‌های صنعتی از اهمیت بسیاری برخوردار هستند. این شبکه‌ها به منظور تامین نیازهایی همچون کنترل، امنیت و نظارت در صنایع استفاده می‌شوند. اگرچه پیدایش این شبکه‌ها فواید فراوانی داشته است اما مشکلاتی نیز به وجود آورده‌اند؛ از جمله‌ی این مشکلات می‌توان به زیاد شدن پیچیدگی، مشکل شدن عیب‌یابی سیستم و از همه مهم‌تر نیاز به امنیت بالا در این سیستم‌ها، اشاره کرد [۱]. در سال‌های اخیر حملات سایبری روی سیستم‌های کنترل صنعتیِ دارای حفظ و پردازش اطلاعات حساس، پیچیده‌تر شده است. زیرساخت‌های بحرانی و ملی اهداف اصلی حملات سایبری است، زیرا اطلاعات یا خدمات اساسی به سیستم‌های آن‌ها بستگی دارد و حمایت از آن‌ها به موضوعی مهم برای سازمان‌ها و کشورها تبدیل شده است. اگر دامنه امنیت سایبری راه‌های کنترل و امیدوار کننده‌ای برای جلوگیری از آن پیدا نکند، بنابر تحلیل‌های موجود تهدیدات سایبری علیه اینترنت‌اشیاء صنعتی تا سال ۲۰۳۰ بالغ بر ۹۰ تریلیون دلار هزینه دربر خواهد داشت [۲]. بنابراین به دلیل اهمیت امنیت در حوزه‌ی اینترنت‌اشیاء صنعتی، در این پژوهش ترافیکِ شبکه‌ی اینترنت‌اشیاء صنعتی مورد بررسی قرار گرفته است. به دلیل اینکه ماهیت شبکه‌های عمومی و شبکه‌های کنترل صنعتی با یکدیگر متفاوت است، نمی‌توان استراتژی‌های تامین امنیت در شبکه‌های عمومی را در شبکه‌های کنترل صنعتی به کار برد. مهم‌ترین موضوع در این زمینه رفتار و مشخصات ترافیک در شبکه‌های کنترل صنعتی می‌باشد. شبکه‌های صنعتی تفاوت زیادی با شبکه‌های تجاری دارند. ازجمله‌ی این تفاوت‌ها می‌توان به تفاوت در ریسک و الویت‌بندی اشاره کرد. به عنوان مثال یک خرابی در شبکه‌های صنعتی می‌تواند مخاطرات بسیار جدی مانند از دست رفتن جان کارکنان یا ضرر مالی وسیع به کشور در پی داشته باشد [۳]؛ همچنین اولویت‌بندی نیازمندی‌های امنیتی در شبکه‌های صنعتی به ترتیب دسترس‌پذیری، یکپارچگی و محرمانگی می‌باشد [۴]. یکی از مهم‌ترین راه‌های تامین امنیت، استفاده از سیستم‌های تشخیص نفوذ است. یکی از تکنیک‌های شناسایی حملات در سیستم‌های تشخیص نفوذ استفاده از یادگیری عمیق می‌باشد، که امروزه به‌طور گسترده در صنایع مختلف استفاده می‌شود.

برای تضمین امنیت، یکی از مکانیزم‌های امنیتی سیستم تشخیص نفوذ می‌باشد؛ نفوذ را تلاشی برای به خطر انداختنِ محرمانگی، یکپارچگی و دسترس‌پذیری (CIA)[3] یا دور زدن مکانیزم‌های امنیتی رایانه یا شبکه توصیف می‌کنند، تشخیص نفوذ یعنی فرآیند نظارت بر وقایع رخ داده در یک سیستم رایانه‌ای یا شبکه‌ای و تجزیه و تحلیل آن‌ها، و سیستم تشخیص نفوذ یک نرم‌افزار یا سیستم سخت‌افزاری برای خودکارسازی فرآیند تشخیص نفوذ است [۵]. همچنین تکنیک‌های یادگیری عمیق به طور فزاینده‌ای برای شناسایی حملات، ناهنجاری‌ها یا نفوذ در یک محیط شبکه‌ی محافظت شده مورد استفاده قرار می‌گیرند. یادگیری عمیق یک عملکرد هوش مصنوعی است که از کارکرد مغز انسان در پردازش داده‌ها و ایجاد الگوهای مورد استفاده در تصمیم‌گیری تقلید می‌کند. از جمله مزیت‌های یادگیری عمیق شامل یادگیری خودکار و چندلایه‌ی ویژگی‌ها در مجموعه داده‌های بزرگ، استخراج خودکارِ نمایش پیچیده‌ی داده‌ها، رویکرد حل مسئله بهتر و همچنین قدرت تعمیم بالا می‌باشد.

مهم‌ترین چالش شبکه‌های عصبی آموزش این شبکه‌ها است و معمولا تعیین فرا پارامترها[4] (وزن هر نورون، تعداد نورون‌ها در هر لایه، تعداد لایه‌ها، نرخ یادگیری و ...) در یک فرآیند تکراری به صورت آزمون و خطا مشخص می‌شود. اغلب پژوهش‌ها تنظیم فرا پارامترها را به صورت دستی انجام داده‌اند و یا اینکه اشاره‌ای به این موضوع در روند پژوهشی خودشان نکرده‌اند، بنابراین می‌توان ساختار شبکه عصبی را با تعیین فرا پارامترهای شبکه عصبی به‌صورت حذف عامل انسانی و خودکارسازی بهینه کرد. یکی از راه‌حل‌های تنظیم فرا پارامترها در این زمینه استفاده از الگوریتم‌های فراابتکاری[5] می‌باشد، که در مقابل دیگر روش‌ها عملکرد بهتری از خود نشان داده‌اند [۶]. بنابراین در این پژوهش به ارائه‌ی رویکردی ترکیبی برای خودکارسازی تنظیم فرا پارامتر در معماری یادگیری عمیق با حذف عامل انسانی پرداخته‌ایم.

در زمینه‌ی الگوریتم‌های فراابتکاری، محققان برای بهبود نقاط ضعف هر یک از الگوریتم‌های فراابتکاری، از ترکیب آن‌ها استفاده می‌نمایند [۷]. برای تعیین فرا پارامترهای شبکه عصبی از الگوریتم‌های فرا ابتکاری بهینه‌سازی ازدحام ذرات (PSO) و وال (WOA) استفاده می‌کنیم. الگوریتم بهینه‌سازی ازدحام ذرات در فاز بهره‌برداری دارای سرعت همگرایی خوب و در فاز اکتشاف دارای محدودیت‌هایی می‌باشد و همچنین الگوریتم بهینه‌سازی وال

[1] Industrial Internet of Things

[2] Industrial Control System

[3] Confidentiality, Integrity, Availability

[4] Hyperparameter

[5] Metaheuristic

قابلیت اکتشاف بسیار خوبی دارد، اما در فاز بهره‌برداری دارای محدودیت‌هایی است؛ بنابراین برای بهبود نقاط ضعف و همچنین بهره‌گیری از مزایای هر دو الگوریتم از ترکیب این دو استفاده می‌کنیم [۶] [۸].

در ادامه ابتدا پیشینه تحقیق (بخش ۲) شرح داده شده، سپس روش پیشنهادی ارائه شده (بخش ۳) و در بخش بعدی ارزیابی و مقایسه با پژوهش‌های پیشین مورد بحث قرار گرفته (بخش ۴) و درنهایت نتیجه‌گیری و کارهای پیشنهادی (بخش ۵) بیان شده است.

۲. مفاهیم پایه

۱,۲ سیستم تشخیص نفوذ

سیستم تشخیص نفوذ کشف‌کننده‌ای است که می‌تواند بسته‌های در حال حرکت از طریق یک یا چند اتصال شبکه را به منظور تشخیص فعالیت مشکوک، تجزیه و تحلیل کند. نقش این سیستم‌ها محدود به هشدار دادن به مدیر سیستم برای ردیابی هرگونه فعالیت غیرعادی در یک میزبان[1] یا شبکه است و مانع از تلاش برای نفوذ نمی‌شود [۹].

۲,۲ سیستم‌های کنترل صنعتی

سیستم‌های کنترل صنعتی[2] (ICSs) بخشی جدایی ناپذیر از زیرساخت‌های مهم هستند و برای مدت طولانی برای نظارت بر ماشین آلات و فرآیندهای صنعتی مورد استفاده قرار می‌گیرند. این سیستم‌ها نظارت و تعامل در زمان واقعی با دستگاه‌ها، جمع‌آوری و تجزیه و تحلیل داده‌ها در زمان واقعی و همچنین ثبت تمام وقایع رخ داده در سیستم‌های صنعتی را انجام می‌دهند.

سیستم کنترل نظارت و جمع‌آوری داده‌ها[3] (SCADA) بزرگ‌ترین زیرمجموعه یک ICS است. استفاده از فناوری IoT در این سیستم‌ها باعث تقویت هوش و امنیت شبکه در بهینه‌سازی و اتوماسیون فرآیندهای صنعتی می‌شود. با توجه به ماهیت حساس بسیاری از کاربردهای صنعتی، امنیت به نگرانی اصلی سیستم‌های SCADA تبدیل شده است. به‌طور خاص، فقدان ملاحظات امنیتی در پروتکل‌های ارتباطی آن‌ها مستقیماً در دسترس بودن، ایمنی و قابلیت اطمینان این سیستم‌ها را به خطر می‌اندازد [۱].

۳,۲ اینترنت‌اشیاء

امروزه اینترنت اشیاء یکی از کلمات کلیدی مورد استفاده در فضای علمی می‌باشد که به طور گسترده مورد بحث محققان و مهندسان قرار گرفته است. اصطلاح اینترنت اشیاء برای اولین بار توسط کوین اشتون[4]، مدیر مرکز Auto-ID در MIT[5] در سال ۱۹۹۹ ارائه شد که بعدا در سال ۲۰۰۵ توسط اتحادیه مخابرات بین المللی[6] معرفی شد. اینترنت اشیاء مفهومی است که اجازه شبکه کردن اشیاء مختلفی از زندگی روزمره و ارتباطات روی اینترنت را، بدون تعامل انسان و با دید بهبود شرایط و روش زندگی می‌دهد.

یکی از حوزه‌های کاربردی اینترنت اشیاء بخش فرآیندهای صنعتی می‌باشد که از آن به عنوان اینترنت اشیاء صنعتی (IIoT) یاد می‌شود. صنایع همواره نگران کاهش مخارج عملیاتی و هزینه‌های مربوطه بوده‌اند؛ بنابراین، شرکت‌ها به طور مداوم در جستجوی راه‌حل‌هایی هستند که باعث بهبود پایداری سیستم، تحمل خطا، انعطاف‌پذیری و کارایی هزینه‌های سیستم می‌شوند؛ با اتخاذ چنین راه‌حل‌هایی پیچیدگی و تعامل ارتباطات درونی سیستم‌های صنعتی گسترش می‌یابد؛ یکی از این راه‌حل‌ها برای رفع نیازهای فعلی سیستم‌های صنعتی، بکارگیری از فناوری اینترنت اشیاء در سیستم‌های کنترل صنعتی است. سیستم‌های کنترل صنعتی عمدتا سیستم‌های حیاتی و با قابلیت دسترسی بالا هستند و عملیات مداوم آن‌ها منجربه تولید حجم عظیمی از داده‌ها می‌شود که از طریق تجزیه و تحلیل داده‌های بزرگ قابل مدیریت هستند همچنین با توجه به ماهیت حساس بسیاری از کاربردهای صنعتی، امنیت به نگرانی اصلی این سیستم‌ها تبدیل شده است. امنیت دستگاه‌های اینترنت اشیاء صنعتی (IIoT) در صورت حمله به دلیل عواقب مخرب بالقوه آن بخصوص در بخش سیستم‌های SCADA بسیار مهم است. اولین حادثه امنیت سایبری در تاریخ سیستم‌های SCADA انفجار خط لوله سیبری در سال ۱۹۸۲ میلادی می‌باشد که در آن یک مهاجم از ویروسی به نام اسب تروجان به عنوان آسیب‌پذیری در سیستم استفاده کرد؛ همچنین در سال ۲۰۱۰ میلادی، سیستم هسته‌ای ایران توسط کرم استاکس‌نت مختل شد [۱۰]. با توجه به اهمیت این حوزه‌ی کاربردی از اینترنت اشیاء، پژوهش ما نیز در این زمینه می‌باشد.

---

[1] Host

[2] Industrial control system

[3] Supervisory control and data acquisition

[4] Kevin Ashton

[5] Massachusetts Institute of Technology(MIT)

[6] International Telecommunication Union(ITU)

۴،۲ یادگیری عمیق

یادگیری عمیق[1] شاخه‌ای از یادگیری ماشین[2] و هوش مصنوعی[3] می‌باشد. هوش مصنوعی یعنی دانش ساخت ماشین‌ها برای کارهایی که برای انجام توسط انسان به هوش نیاز دارد؛ در واقع هدف آن شبیه‌سازی و درک رفتار انسان می‌باشد؛ کاربردهای آن شامل رباتیک، درک گفتار، بازی‌های کامپیوتری، اقتصاد و رفتارشناسی و ... می‌باشد. مهم‌ترین ابزار هوش مصنوعی یادگیری عمیق می‌باشد. یادگیری ماشین زیرمجموعه‌ای از هوش مصنوعی است که می‌تواند کارهای غیرممکن یا بسیار دست و پاگیر را با زبان‌های برنامه‌نویسی سنتی حل کند [۱۱].

۳. پیشینه تحقیق

در پژوهش TR-IDS یک روش تشخیص نفوذ مبتنی بر ناهنجاری را از طریق شبکه عصبی کانولوشنِ متنی[4] و جنگل تصادفی[5] در دستگاه‌های اینترنت اشیاء ارائه می‌دهد [۱۲]. با توجه به استفاده‌ی بیشتر روش‌های تشخیص از اطلاعات موجود در سرآیند بسته‌ها یا اطلاعات آماری مربوط به کل جریان، قادر به شناسایی محتوای مورد سوء استفاده در متن بسته‌ها نیستند. در این پژوهش یک سیستم جدید برای تشخیص نفوذ به نام TR-IDS پیشنهاد می‌شود که ویژگی‌های آماری و سرآیند بسته‌ها و همچنین ویژگی‌های متن بسته‌ها را در نظر می‌گیرد. ویژگی‌های آماری شامل فیلدهای موجود در سرآیند بسته‌ها و ویژگی‌های آماری کل جریان به صورت دستی از هر جریان عبوری از شبکه استخراج می‌شود. ویژگی‌های متن بسته‌ها با استفاده از تعبیه کلمه به یک بردار کلمه به وسیله‌ی الگوریتم Skip-gram ترسیم شده و سپس با استفاده از شبکه عصبی کانولوشنِ متنی ویژگی‌های برجسته متن بسته‌ها استخراج می‌شود. با ادغام ویژگی‌های آماری و متن بسته‌ها از الگوریتم جنگل تصادفی برای طبقه‌بندی مجموعه داده‌ی جدید استفاده می‌شود. الگوریتم جنگل تصادفی در داده‌های ساخت‌یافته عملکرد بهتری دارد و شبکه عصبی کانولوشن برای مدیریت داده‌های بدون ساختار مناسب است، که در این پژوهش مطابق با اهداف مسئله از هر دو الگوریتم استفاده شده است. پارامترهای ارزیابی این پژوهش با استفاده از مجموعه داده‌های ISCX2012 که توسط مرکز امنیت اطلاعات عالی (ISCX) دانشگاه نیوبراتزویک (UNB) تولید شده، به درستی ۹۹،۱۳ درصد، نرخ تشخیص ۹۹،۲۶ درصد و نرخ هشدار اشتباه ۱،۱۸ درصد دست پیدا می‌کنند. از جمله معایب این پژوهش استخراج ویژگی‌های آماری به صورت دستی می‌باشد که با استفاده از روش‌های متنوع موجود می‌توان ویژگی‌های آماری را به صورت خودکار استخراج کرد، همچنین در مورد تنظیم فرا پارامترهای شبکه‌ی عصبی بحث نشده است.

در پژوهش DAE-DFFNN به شناسایی فعالیت‌های مخرب در اینترنت اشیاء صنعتی برای پیشگیری از نرخ مثبت کاذب بالا و افزایش نرخ تشخیص[6] می‌پردازد [۲]. در این پژوهش از معماری‌های مختلف مدل یادگیری عمیق برای توسعه یک سیستم تشخیص ناهنجاری برای سیستم‌های کنترل صنعتی اینترنتی استفاده می‌نماید. در مرحله آموزش، یک الگوریتم خودرمزنگار عمیق[7] با استفاده از مشاهدات شبکه‌ای نرمال برای ایجاد پارامترهای اولیه آموزش داده می‌شود. پارامترهای اولیه شامل وزن و بایاس بوده و این الگوریتم یک بازنمایی عمیق از مشاهدات نرمال را می‌آموزد. این پارامترها به عنوان یک مرحله اولیه برای آموزش یک شبکه عصبی پیش‌خور عمیق[8] (DFFNN) برای کشف نمونه حملات موجود و حملات جدید مورد استفاده قرار می‌گیرد. گره‌های مختلف پنهان در این تکنیک به صورت حرفه‌ای بازنمایی ویژگی‌های عمیق را یادگرفته و مهمترین ویژگی‌ها را با تبدیل ابعاد بالای داده به ابعاد پایین مبتنی بر کاهش لایه پنهان دریافت می‌نمایند. ساختار و استقرار سیستم تشخیص ناهنجاری به ترتیب شامل واحد شنود و مانیتورینگ، پایگاه داده و واحد تحلیل و پاسخ در گیت‌وی اینترنت اشیاء می‌باشد. با دو مجموعه داده شبکه‌ی NSL-KDD و UNSW-NB 15 مورد ارزیابی قرار گرفته و نتایج آزمایشات تجربی برای مجموعه‌داده‌ها به ترتیب شامل درستی ۹۸،۶ و ۹۲،۴ درصد، نرخ تشخیص ۹۹ و ۹۳ درصد و نرخ هشدار اشتباه ۱،۸ و ۸،۲ درصد دست پیدا کرده‌اند. در این پژوهش تنها به تنظیم پارامترهای اولیه‌ی وزن‌ها و بایاس‌ها و نیز کاهش ابعاد داده از طریق الگوریتم خود رمزنگار عمیق پرداخته شده و در مورد دیگر پارامترهای قابل تنظیم در شبکه‌ی یادگیری عمیق بحثی نشده است، همچنین به نتایج ارزیابیِ قابل قبولی در مجموعه داده‌ی UNSW-NB15 دست پیدا نکرده‌اند.

[1] Deep Learning
[2] Machine Learning
[3] Artificial Intelligence
[4] Text-Convolutional Neural Network
[5] Random Forest
[6] Detection rate
[7] Deep auto-encoder
[8] Feedforward Neural Network

در پژوهش گونزالـو د لا تـوره پـارا و همکـاران بـه تشـخیص حمـلات اینترنت‌اشیاء با استفاده از یادگیری عمیق توزیع‌شده مـی‌پـردازد [۱۳]. این تحقیق بر شناسایی حملاتِ سطح دستگاه در سـمت کلاینـت و یـا در قسمت میزبان back-end روی ابر به‌طور همزمان، متمرکـز اسـت. علاوه براین چهارچوب پیشنهادی شامل شناسایی و دفاع در نقطه مبـدا حمله به وسیله‌ی جاسازیِ مدل شـبکه عصـبی کانولوشـن (CNN) در دستگاه کلاینت است؛ همچنین مدل مبتنـی بـر حافظـه کوتـاه مـدت (LSTM) در میزبان back-end در راستای مدل قبلی برای شناسـایی حملات توزیع شده و باتنت‌ها و نیز حمـلات دارای تجزیـه و تحلیـل و محاسبات بیشتر، اسـتفاده مـی‌شـود. ایـن پـژوهش شـامل یـک روش آموزش مشترک برای به حداقل رساندن استفاده منابع در دستگاه‌هـای اینترنت اشیاء و به حداکثر رساندن سـودمندی ویژگـی‌هـای اسـتخراج شده در سمت سرور back-end می‌باشد. در آخر امکان ادغام خودکار چندین درخواست URL بـرای بهبـود و عملکـرد کلی(دقـت و تحمـل خطا) در سیستم فـراهم مـی‌شـود. بـرای آمـوزش و آزمـایش و اعتبـار سنجی مدل شبکه عصبی از مجموعه داده N-BaIoT اسـتفاده شـده و نیـز یـک مجموعـه داده متشـکل از URLهـای فیشـینگ شـامل PhishTank(OpenDNS,2016)و OpenPhish(OpenPhish) و نیز غیر فیشینگ شامل Curlie(Curlie,2018) می‌باشد. در نهایـت به وسیله‌ی مدل CNN در سمت دسـتگاه‌هـای کلاینـت بـه شناسـایی حملات فیشینگ با درستی ۹۴،۳ درصد و نیز در مـدل LSTM سـمت میزبان back-end به شناسـایی حمـلات بـاتنـت بـا درسـتی ۹۴،۸ درصد دست پیدا کـرده‌انـد. در ایـن پـژوهش تنظـیم فـرا پارامترهـای ساختار شبکه‌ی عصبی در هـر دو بخـش CNN و LSTM بـه صـورت دستی انجام شده است.

در پژوهش DBN-PNN یک روشی برای تشخیص نفوذ از طریق شبکه باور عمیق[1] و شبکه عصبی احتمالی[2] ارائه داده‌اند [۱۴]. ژائو و همکارانش برای مشکلاتی از قبیل مقدار زیاد داده‌ها، طولانی بودن زمان آموزش و گیرکردن در مینیمم‌های محلی به روشی ترکیبی مبتنی بر شبکه عصبی پرداخته‌اند. چهارچوب کلی در این پژوهش بدین صورت است که ابتدا یک پیش‌پردازش و نرمال‌سازی روی مجموعه داده‌ی KDD CUP 1999 انجام شده است که تعداد ویژگی‌های بهینه انتخاب می‌شود؛ بعد از آن شبکه عصبی DBN برای یادگیری ویژگی‌ها و پیش آموزش مورد استفاده قرار گرفته است. برای تعیین ساختار این شبکه عصبی از الگوریتم بهینه‌سازی توده ذرات[3] (PSO) برای بهینه کردن تعداد نورون‌های لایه مخفی در هر لایه شبکه عصبی استفاده کرده‌اند. در نهایت برای کلاس‌بندی داده‌ها از شبکه عصبی احتمالی استفاده شده است. با توجه به روش ارائه شده به نرخ تشخیص ۹۳ درصد دست یافته‌اند. از معایب این مدل نرخ تشخیص تقریبا پایین و همچنین استفاده از مجموعه داده‌ای قدیمی می‌توان اشاره کرد.

در پژوهش هوان‌یانگ و همکارانش به تشخیص نفوذ شـبکه مبتنـی بـر یادگیری عمیق برای سیستم‌های کنترل نظارت و جمـع‌آوری داده‌هـا[4] (SCADA) پرداخته‌اند [۱۵]. رویکرد پیشنهادی بـدون قطـع عملکـرد عادی سیستم و به صورت آنلاین و بـا اسـتفاده از قابلیـت مانیتورینـگ دستگاه‌های موجود در شبکه SCADA انجام شده است. تشـخیص را از طریق یک شبکه عصبی کانولوشنی (CNN) برای توصـیف الگـوه‌هـای زمانی و برجسته‌ی رفتارهای شبکه در میزبـان SCADA و بـه‌وسـیله‌ی جمع‌آوری بسته‌هـای ترافیـک در سـوئیچ‌هـای شـبکه ارائـه مـی‌دهـد. شناسایی حملات پیچیده و واقعی برای پروتکل‌های شـبکه‌ی SCADA در ابتدایی‌ترین حالت ممکـن انجـام مـی‌شـود. همچنـین یـک برنامـه آموزش مجدد با استفاده از خوشه‌بندی K میانه برای شناسایی و اصلاح حملات اختصاصی شبکه و نیز سازگاری با محـیط‌هـای مختلـف ارائـه می‌دهد. ارزیابی در مجموعـه داده‌هـای جمـع‌آوری شـده از بسـترهای آزمون سیستم‌های انتقال انرژی مختلـف بـا تجهیـزات میـدانی واقعـی انجام می‌شود. در نهایت، در آزمایشات اولیه و ثانویه به ترتیب بـه دقـت کلـی تشـخیص ۹۹،۳۸ درصـد و ۹۹،۸۴ درصـد رسـیده‌انـد. از جملـه مزایـای ایـن روش اسـتفاده از مجموعـه ترافیـک واقعـی SCADA در سیستم‌های انتقال انرژی نام برد. برای بهبود می‌توان با استفاده از دیگـر مجموعه‌داده‌های واقعی یا مجموعه داده‌های در دسترس کـارایی مـدل را بررسی کرد و نیز ساختار و فرا پارامترهای شبکه عصـبی کانولوشـنی را با استفاده از الگوریتم‌های فرا ابتکاری یا دیگر روش‌ها خودکار سـازی کرد.

در پژوهش Intelligent-IDS یک سیستم تشخیص نفوذ هوشمند با استفاده از یادگیری عمیق ارائه شده است [۱۶]. یک

[1] Deep Belief Network

[2] Probabilistic Neural Network

[3] Particle Swarm Optimization

[4] Supervisory Control and Data Acquisition

روش یادگیری عمیق برای توسعه‌ی یک سیستم تشخیص نفوذ انعطاف پذیر و مؤثر برای شناسایی و طبقه‌بندی حملات سایبریِ پیش‌بینی نشده، استفاده شده است. از طریق ترکیب روش‌های تشخیص مبتنی بر امضاء و ناهنجاری برای شناسایی حملات سایبری با استفاده از یک رویکرد شبکه عصبی عمیق DNN پیشنهاد شده است. در این پژوهش فرا پارامترهای شبکه عصبی و توپولوژی شبکه‌ی بهینه برای شبکه عصبی DNN، به صورت دستی و با دوره آموزشی ۱۰۰۰ دور و محدوده نرخ یادگیری عمیق ۰,۵ تا ۰,۰۱ انجام شده است. نقاط ضعف روش پیشنهادی بهینه‌سازی فرا پارامترهای شبکه عصبی به صورت دستی و از جمله مزایای آن استفاده از مجموعه داده‌های متعدد از جمله UNSW-NB15، NSL-KDD، CICIDS2017، WSD-DS، KDDCup99 می‌باشد. نرخ درستی مجموعه داده NSL-KDD در محدوده ۹۵ تا ۹۹ درصد و برای مجموعه داده UNSW-NB15 در محدوده ۶۵ تا ۷۵ درصد می‌باشد.

در پژوهش HDRaNN یک شبکه عصبی تصادفی عمیقِ ترکیبی برای تشخیص حملات سایبری در اینترنت‌اشیاء صنعتی ارائه شده است [۱۷]. در این پژوهش از یک شبکه عصبی تصادفی عمیق (DRaNN) و یک پرسپترون چند لایه [۱] (MLP) تشکیل شده است. این مدل شامل یک لایه ورودی، سه لایه RNN ، سه لایه MLP و یک لایه خروجی است. در معماری پیشنهادی برای جلوگیری از بیش‌برازش از لایه‌های dropout استفاده شده است. تعداد دوره آموزشی ۱۵۰ دوره آموزشی در نظر گرفته شده و مقدار بهینه برای فرا پارامتر نرخ یادگیری با آزمون و خطا به دست آمده است. برای آزمایش از دو مجموعه داده‌ی DS2OS و UNSW-NB15 استفاده شده که در هر دو مجموعه داده به درستی ۹۸ درصد دست پیدا کرده‌اند. از جمله معایت این روش تعیین فرا پارامترهای شبکه عصبی به صورت دستی و همچنین از مجموعه داده‌های آموزش و آزمایش مخصوص خودِ مجموعه‌داده‌ها استفاده نشده است، که این معایب منجربه سیستم تشخیص نفوذ غیر قابل اتکاء می‌شود. از جمله مزایای این روش می‌توان به نرخ درستی بالای آن اشاره کرد.

پژوهش‌های C-LSTM [۱۸]، RNN-IDS [۱۹] و SDN-DNN [۲۰] یک سیستم تشخیص نفوذ مبتنی بر شبکه‌های عصبی ارائه داده‌اند. یکی از نقاط ضعف اصلی این پژوهش‌ها نیز تنظیم فراپارامترهای شبکه عصبی به صورت دستی می‌باشد، در مورد این پژوهش‌ها به‌صورت جزئی‌تر در جدول ۱ توضیح داده شده است.

[1] Multilayer perceptron

جدول ۱ مقایسه کارهای انجام شده

پژوهش	روش پیشنهادی	نقاط قوت و ضعف
TR-IDS [12] **سال انتشار ۲۰۱۸**	استفاده از ویژگی‌های سرآیند بسته‌ها و ویژگی‌های متن بسته‌ها در تشخیص نفوذ با استفاده از شبکه عصبی کانولوشن و جنگل تصادفی	**نقاط قوت:** در نظر گرفتن ویژگی‌های متن بسته‌ها **نقاط ضعف:** استخراج ویژگی‌های آماری و تنظیم فرا پارامترهای شبکه عصبی به صورت دستی
C-LSTM [18] **سال انتشار ۲۰۱۸**	تشخیص ناهنجاری ترافیک وب با استفاده از CNN و LSTM	**نقاط قوت:** دقت مناسب و استفاده از مدل پیشنهادی مناسب با توجه به ساختار مجموعه داده **نقاط ضعف:** تنظیم فرا پارامترهای شبکه عصبی به صورت دستی
DAE-DFFNN [2] **سال انتشار ۲۰۱۸**	شناسایی فعالیت‌های مخرب در اینترنت‌اشیاء صنعتی با استفاده از الگوریتم خودرمزنگار عمیق و شبکه عصبی پیش‌خور عمیق	**نقاط قوت:** تنظیم پارامترهای وزن و بایاس و کاهش ابعاد داده **نقاط ضعف:** نتایج ارزیابی ضعیف در مجموعه داده UNSW-NB 15 - استفاده‌ی نامناسب از داده‌های آموزش و آزمایش
گونزالو د لا توره پارا و همکاران [۱۳] **سال انتشار ۲۰۲۰**	تشخیص حملات اینترنت‌اشیاء صنعتی با استفاده از CNN و LSTM	**نقاط قوت:** مدل پیشنهادی مناسب برای به حداقل رساندن استفاده از منابع در IoT **نقاط ضعف:** تنظیم فرا پارامترهای شبکه عصبی به صورت دستی
DBN-PNN [14] **سال انتشار ۲۰۱۷**	روشی برای تشخیص نفوذ از طریق شبکه باور عمیق و شبکه عصبی احتمالی	**نقاط قوت:** تنظیم فرا پارامتر با استفاده از PSO **نقاط ضعف:** نرخ تشخیص تقریبا پایین و استفاده از مجموعه داده قدیمی
هوان‌یانگ و همکارانش [۱۵] **سال انتشار ۲۰۱۹**	تشخیص نفوذ در سیستم‌های اسکادا با استفاده از CNN	**نقاط قوت:** استفاده از مجموعه ترافیک واقعی اسکادا **نقاط ضعف:** تنظیم فرا پارامترهای شبکه عصبی به صورت دستی
RNN-IDS [19] **سال انتشار ۲۰۱۷**	ارائه سیستم تشخیص نفوذ برای شبکه عصبی بازگشتی	**نقاط قوت:** تشخیص در زمان آموزش و آزمایش پایین **نقاط ضعف:** تعیین فرا پارامتر به صورت آزمون و خطا، نرخ درستی پایین
SDN-DNN [20] **سال انتشار ۲۰۱۶**	سیستم تشخیص نفوذ در شبکه SDN با استفاده از DNN	**نقاط قوت:** پیاده‌سازی در بستر معماری جدید شبکه **نقاط ضعف:** تعیین فرا پارامتر به صورت آزمون و خطا

		با نرخ درستی نسبتا پایین
Intelligent-IDS [۱۶] **سال انتشار ۲۰۱۹**	سیستم تشخیص نفوذ هوشمند با استفاده از یادگیری عمیق	**نقاط قوت:** استفاده از روش تشخیص ترکیبی و آزمایشات گسترده در چندین مجموعه داده **نقاط ضعف:** تنظیم فرا پارامتر به صورت آزمون و خطا
HDRaNN [17]	ارائه‌ی یک شبکه عصبی تصادفی عمیقِ ترکیبی برای تشخیص حملات سایبری در اینترنت‌اشیاء صنعتی	**نقاط قوت:** نرخ درستی بالا **نقاط ضعف:** تنظیم فرا پارامترها به صورت آزمون و خطا – استفاده‌ی نامناسب از داده‌های آموزش و آزمایش

با توجه به بررسی کارهای انجام شده در حوزه‌ی تشخیص نفوذ، بیشتر روش‌هایی که مورد بررسی قرار گرفته، روش‌های مبتنی بر شبکه بودند. در این پژوهش‌ها از روش‌های هوش مصنوعی و شبکه‌های عصبی به همراه الگوریتم‌های یادگیری ماشین استفاده شده است. نتایجی که از اجرای این روش‌ها به دست آمده نشان دهنده کارآمدی و اثربخشی این روش‌ها می‌باشد. کارهای انجام شده بیشتر به استفاده از شبکه عصبی اکتفا کرده‌اند؛ یکی از نقاط ضعف این روش‌ها بهینه‌سازی فرا پارامترهای شبکه عصبی به صورت دستی بوده، و بهینه بودن ساختار شبکه عصبی در میزان دقت و زمان آموزش روش‌های ارائه شده مؤثر خواهد بود. در جدول ۱ روش‌های مورد مطالعه و نقاط قوت و ضعف هر کدام مورد بررسی قرار گرفته است.

۴. روش پیشنهادی

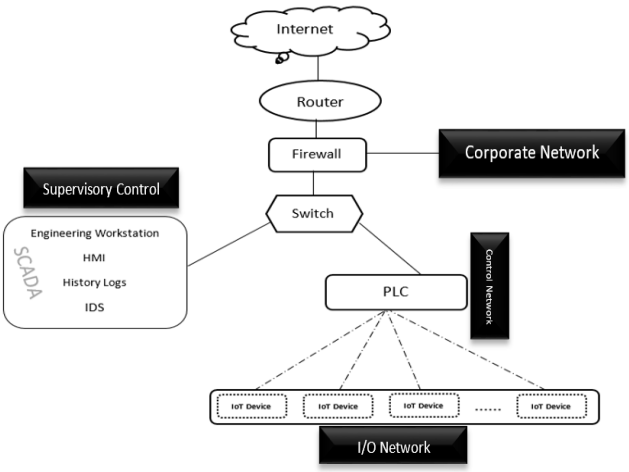
با توجه به پژوهش‌هایی که در فصل سوم بررسی شد یکی از مهم‌ترین ابزارهای مورد استفاده برای سیستم‌های تشخیص نفوذ در حوزه‌ی اینترنت‌اشیاء صنعتی، شبکه‌های عصبی هستند. مهم‌ترین چالش آموزش شبکه‌های عصبی تنظیم فرا پارامترهای اولیه در این شبکه‌هاست.

ما دراین پژوهش برای ایجاد یک سیستم تشخیص نفوذ از ترکیب الگوریتم‌های شبکه عصبی کانولوشن (CNN) و شبکه عصبی بازگشتی مبتنی بر حافظه کوتاه مدت (LSTM) استفاده کرده‌ایم. با انتخاب شبکه عصبی CNN-LSTM و ایجاد ساختاری مناسب برای این شبکه عصبی، گام موثری در ایجاد یک روش سیستم تشخیص نفوذ هم در فاز آموزش و هم در فاز آزمایش برداشته‌ایم. اما به‌صرف ایجاد این چهارچوب براساس شبکه عصبی CNN-LSTM، یک سیستم تشخیص نفوذ قابل اطمینان و اتکاء نخواهیم داشت، چراکه عملکرد و کارایی شبکه عصبی منوط به تعیین فرا پارامترهایی در شبکه عصبی است، که باید به صورت آزمون و خطا توسط مدیر سیستم تعیین شود. تعداد زیاد آزمون و خطا با توجه به مقدارهایی که به صورت تصادفی توسط مدیر انتخاب می‌شود، یکی از چالش‌هایی است که کارایی، عملکرد و دقت تشخیصِ شبکه عصبی را تحت تاثیر قرار داده است. استفاده از ترکیب الگوریتم پویای بهینه‌سازی ازدحام ذرات (PSO) و الگوریتم بهینه‌سازی وال (WOA) که دارای عملکرد مناسب و سرعت همگرایی بالا در فاز اکتشاف و بهره‌برداری هستند [۲۱] [۲۲]، راه‌حل خوبی برای چالش ذکر شده می‌باشد. بنابراین در این پژوهش برای تنظیم فرا پارامترهای شبکه عصبی از ترکیب الگوریتم‌های فرا ابتکاری ازدحام ذرات و وال (PSO-WOA) استفاده شده است، که در جهت بهبود کارایی و تعیین فرا پارامترهای بهینه در سیستم تشخیص نفوذ مبتنی بر شبکه عصبی CNN-LSTM در فاز آموزش می‌باشد. در نهایت، روش پیشنهادی شامل شبکه عصبی CNN-LSTM و الگوریتم‌های فرااِبتکاری بهینه‌سازی ازدحام ذرات و وال (PSO-WOA) را روش PWCL[1] می‌نامیم که در ادامه به معرفی دقیق‌تر این روش خواهیم پرداخت.

در شکل ۱ نمای کلی از معماری سیستم‌های اسکادا نشان داده شده است. معمولا سیستم‌های اسکادا از چهار زیر سیستم مختلف تشکیل شده‌اند، شبکه I/O[2]، کنترل نظارت[3]، شبکه کنترل[4] و شبکه گروهی[5]. شبکه I/O از دستگاه‌های مستقر IIoT (شامل حسگرها و محرک‌ها) در فرآیند صنعتی تشکیل شده است. زیرسیستم اصلی کنترل نظارتی مسئول تامین امنیت، کنترل و نظارت بر دستگاه‌های IIoT است و سیستم تشخیص نفوذ نیز در این قسمت قرار می‌گیرد. شبکه کنترل شامل کنترل‌کننده‌های منطقی قابل برنامه‌ریزی (PLC) است که به‌طور مستقیم

[1] PSO-WAL and CNN-LSTM
[2] Input/Output
[3] Supervisory control
[4] Control Network
[5] Corporate Network

فرآیندهای فیزیکی را حس و مدیریت می‌کنند. از آن‌جا که سنسورها با محرک‌ها نمی‌توانند مستقیما ارتباط برقرار کنند، از PCL برای جمع‌آوری داده‌های حس شده و ارسال دستورات به محرک‌ها استفاده می‌شود. سرانجام، شبکه گروهی متشکل از سرورها، رایانه‌ها و سایر کاربران متصل به شبکه برای سایر خدمات عمومی مانند انتقال پرونده، میزبانی وب سایت، سرورهای پستی، برنامه‌ریزی منابع و غیره است [۱].



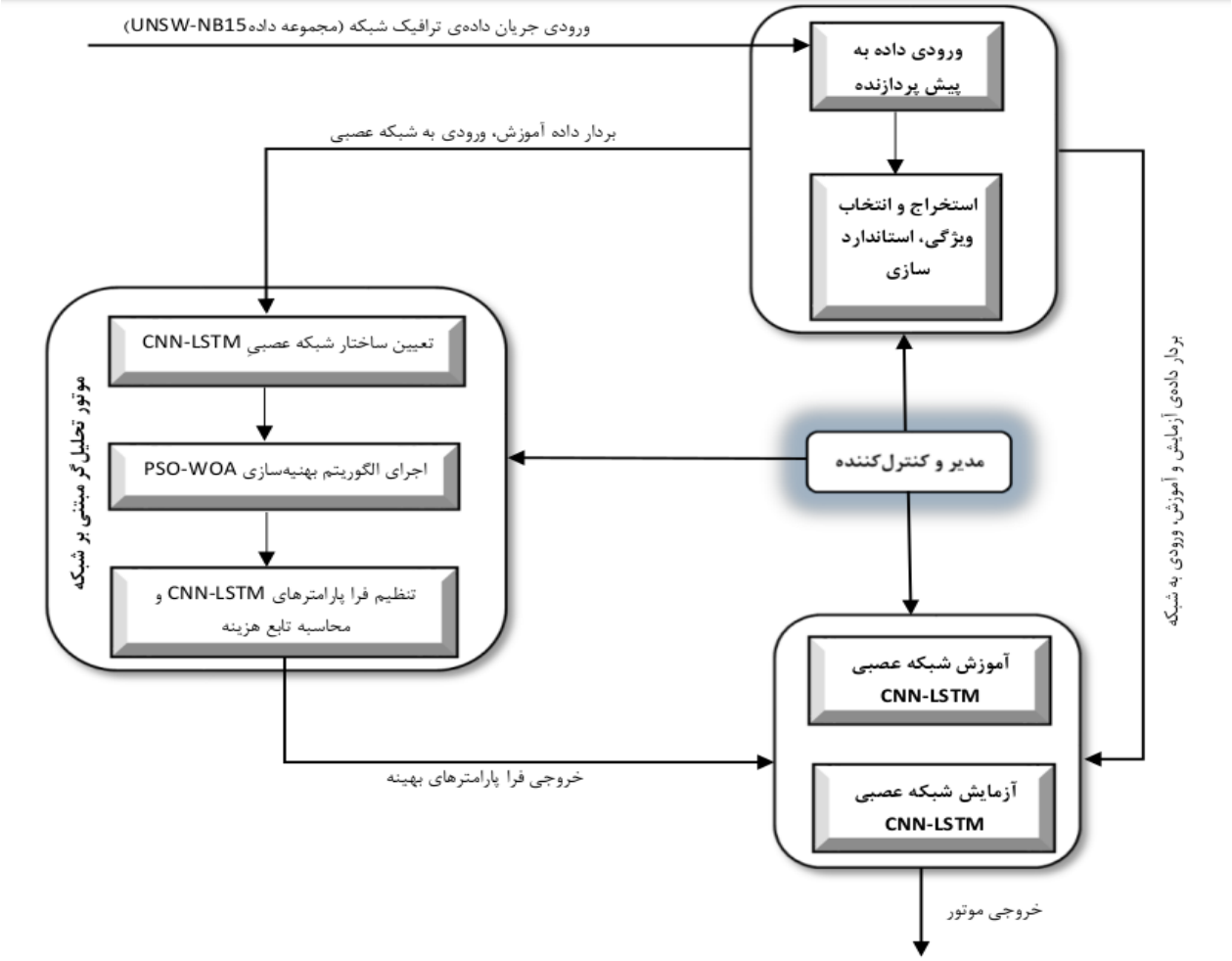
شکل ۱ قرارگیری سیستم تشخیص نفوذ در زیر سیستم اصلی کنترل نظارتی در شبکه‌ی اینترنت‌اشیاء صنعتی [۱]

دلیل اصلی انتخاب شبکه عصبی کانولوشن و شبکه عصبی بازگشتی مبتنی بر حافظه کوتاه مدت برای موتور تحلیل‌گر سیستم تشخیص نفوذ، ویژگی‌های متفاوت و خوب هر دو روش می‌باشد. شبکه‌های عصبی کانولوشن با کاهش اتصالات بین لایه‌ها، مقیاس‌پذیری را افزایش داده و پیچیدگی زمان آموزش را بهبود می‌بخشد، همچنین به دلیل یادگیری خودکار ویژگی‌ها از داده‌های خام، در رویکردهای آموزشِ یادگیری عمیق مورد استفاده قرار می‌گیرد. شبکه‌های عصبی بازگشتی مبتنی بر حافظه کوتاه مدت به دلیل توانایی این شیوه در مدیریت موثرِ داده‌های متوالی مورد استفاده قرار می‌گیرد. این قابلیت برای کارهای مختلف مانند تشخیص تهدید با الگوهای تهدید وابسته به زمان، سودمند است. بنابراین، استفاده از اتصالات مکرر می‌تواند شبکه‌های عصبی را بهبود بخشد و الگوهای مهم رفتاری را استخراج کند [۲۳]. با استفاده از ترکیب هر دو شبکه‌ی عصبی به نام CNN-LSTM، ما توانسته‌ایم به خوبی از مزیت‌های هر دو روش استفاده کنیم. در شبکه‌های عصبی بانظارت، تعیین ساختار بهینه و مناسب در کارایی شبکه عصبی به خصوص در افزایش نرخ تشخیص و کاهش زمان آموزش بسیار حائز اهمیت است، زیرا عمدتا یکی از چالش‌های سیستم تشخیص نفوذ پیچیدگی محاسباتی و زمان‌بر برای فرآیند آموزش در مدل می‌باشد، لذا بایستی مدلی برای کاهش پیچیدگی و کاهش تعداد تکرارهای آزمون و خطا(برای یافتن شبکه عصبی با ساختار و فرا پارامترهای بهینه) ایجاد کنیم.

مطابق با شکل ۲ در روش PWCL، دو فاز متفاوت وجود دارد، فاز اول مربوط به آموزش و فاز دوم آزمایشِ سیستم تشخیص نفوذ می‌باشد. که با توجه به توضیحات داده شده در مورد شبکه‌های عصبی یک سیستم تشخیص نفوذ مبتنی بر شبکه عصبی CNN-LSTM پیشنهاد کردیم. در موتور تحلیل‌گر روش PWCL، شبکه عصبی CNN-LSTM، در دو فاز آموزش و آزمایش مورد استفاده قرار گرفته است، در هر دو فاز آموزش و آزمایش فرا پارامترهایی توسط مدیر تعیین می‌شود، مقادیر این فرا پارامترها در نرخ تشخیص و دقت سیستم بسیار موثر است، بنابراین برای تعیین این فرا پارامترها باید به تعداد دفعات زیادی شبکه را آموزش داده تا اینکه به یک شبکه‌ای با ساختار بهینه برسیم به‌طوری که اهداف را برآورده سازد.

برای تعیین فرا پارامترهای شبکه عصبی از الگوریتم‌های فرا ابتکاری بهینه‌سازی ازدحام ذرات (PSO) و وال (WOA) استفاده می‌کنیم. الگوریتم بهینه‌سازی ازدحام ذرات در فاز بهره‌برداری دارای سرعت همگرایی خوب و در فاز اکتشاف دارای محدودیت‌هایی می‌باشد و همچنین الگوریتم بهینه‌سازی وال قابلیت اکتشاف بسیار خوبی دارد، اما در فاز بهره‌برداری دارای محدودیت‌هایی است؛ بنابراین برای بهبود نقاط ضعف و همچنین بهره‌گیری از مزایای هر دو الگوریتم از ترکیب این دو استفاده می‌کنیم.

روش PWCL متشکل از ۳ مؤلفه اصلی است. در شکل ۲ شمای کلی روش PWCL ترسیم شده است. همان‌طور که در شکل نمایش داده شده است، در بین این ۳ مؤلفه، مؤلفه موتور تحلیل‌گر از اهمیت ویژه و بسزایی برخوردار است. فرا پارامترها و مشخصات



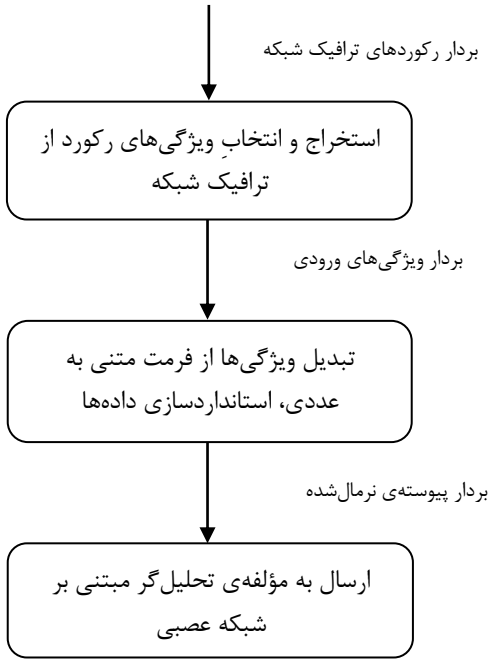
شکل ۲ معماری روش PWCL

هریک از ۳ مؤلفه‌ی اصلی که در ادامه تشریح شده است، توسط کاربر سیستم قابل تنظیم می‌باشند. این مؤلفه‌ها عبارت‌اند از:

۱. مؤلفه پیش‌پردازنده[1]
۲. مؤلفه موتور تحلیل‌گر مبتنی بر شبکه عصبی CNN-LSTM [2]
۳. مؤلفه مدیر و کنترل‌کننده[3]

۱,۴ مؤلفه پیش‌پردازنده

در یک محیط واقعی مطابق شکل ۳، ابتدا استخراج و انتخاب ویژگی‌های رکورد از ترافیک شبکه صورت می‌پذیرد؛ که در این پژوهش این مرحله را از مجموعه داده‌ی UNSW-NB15 [۲۴] تحویل می‌گیریم. از این مجموعه داده برای توسعه‌ی تشخیص نفوذ در سیستم‌های اینترنت‌اشیاء صنعتی مورد استفاده قرار می‌گیرد. سپس در این شبیه‌سازی مراحل تبدیل ویژگی‌ها به فرمت عددی، استانداردسازی و انتخاب ویژگی‌های ویژه انجام شده و داده‌ها آماده‌ی ورود به نورون‌های لایه‌ی ورودیِ شبکه عصبی در بخش موتور تحلیل‌گرِ مبتنی بر شبکه عصبی می‌باشد.



شکل ۳ روند پیش پردازش

[1] Data Preprocessor
[2] Neural Network CNN-LSTM based Analyzer
[3] Manager and Controller

با توجه به مطالب گفته شده، به طور کلی فرآیند پیش‌پردازش در این پژوهش را می‌توان به سه مرحله تقسیم کرد:

۱) تبدیل فرمت متنی داده‌ها به فرمت عددی : داده‌های ورودی در مجموعه‌داده‌ی مورد نظر به صورت جدولی و شامل متغیرهای اسمی[1] ، عدد صحیح[2] ، وابسته به زمان[3] و دودویی[4] است که در نهایت به دو فرمت عددی و متنی دسته‌بندی می‌شوند. اولین گام در آماده‌سازی داده‌ها، تبدیل ویژگی‌های متنی به ویژگی‌های عددی با استفاده از رمزگذاری برچسب[5] در طول فرآیند است.

۲) استانداردسازی : استاندارد سازی داده کمک می‌کند که اهمیت آن‌ها به واحد اندازه‌گیری‌شان بستگی نداشته باشد. در نتیجه در مواردی مانند داده‌کاوی و تحلیل داده‌های چند متغیره از داده‌های استاندارد شده استفاده می‌شود. استانداردسازی میانگین داده‌ها را به صفر و واریانس[6] را به یک تبدیل می‌کند. هرچه مقادیر داده به سمت بی‌نهایت کشیده شود، واریانس داده‌ها به یک نزدیک‌تر می‌شود. فرمول تابع استانداردسازی یا نمره استاندارد[7] مطابق با فرمول (۱) محاسبه می‌شود [۲۵].

$$Z_i = \frac{X_i - \mu}{\sigma} \quad (۱)$$

۳) انتخاب ویژگی‌های ویژه : روش‌های زیادی برای کاهش تعداد ویژگی وجود دارد. انتخاب زیرمجموعه‌ای از ویژگی‌ها، که تا حد زیادی مانند مجموعه کامل ویژگی‌ها رفتار می‌کنند و باعث کاهش پیچیدگی زمانی و محاسباتی سیستم در هر دو فاز آموزش و آزمایش می‌شود. در این پژوهش برای انتخاب ویژگی‌های ویژه از روش تحلیل مؤلفه‌های اصلی (PCA) استفاده شده است [۲۶]. روش PCA برای انتخاب ویژگی با تعیین مؤلفه‌های اصلی یا (PC)ها در داده‌ها این کار را انجام می‌دهد. مؤلفه‌های اصلی در حقیقت همان بردار ویژه‌های ماتریس کوواریانس داده‌ها هستند. بیشترین واریانس داده‌ها در راستایی قرار دارد که بردار ویژه‌ی متناظر با بزرگترین مقدار ویژه در آن راستا قرار دارد. در این پژوهش از واریانس تجمعی در تکنیک تحلیل مؤلفه‌های اصلی (PCA) استفاده شده است. واریانس تجمعی نشان می‌دهد که چند درصد از اطلاعات موجود در مؤلفه‌ها به وسیله‌ی تعداد مشخص از مؤلفه‌ها قابل ارائه است.

یک گام مهم در آموزش، انتخاب و استخراج ویژگی از ترافیک در شبکه‌های مورد بررسی می‌باشد. در پژوهش ذوالانواری و همکاران [۱] با ایجاد یک بستر تست واقعی در شبکه‌ی اینترنت‌اشیاء صنعتی و همچنین پیاده‌سازی سناریوهای واقعیِ اینترنت‌اشیاء صنعتی، یک سیستم تشخیص نفوذ قابل اتکایی را ارائه داده‌اند. در این بستر آزمایشی از پروتکل Modbus، یکی از محبوب‌ترین پروتکل‌های اینترنت‌اشیاء صنعتی، استفاده شده است. به‌وسیله‌ی انتخاب و استخراج ویژگی‌های بالقوه، اهمیت ویژگی‌های مختلف در تشخیص ترافیک عادی از ترافیک حمله در شبکه‌ی اینترنت‌اشیاء صنعتی بررسی شده و در نهایت، ۲۳ ویژگی مهم در نظر گرفته شده است. با توجه به اینکه اکثر ویژگی‌های پژوهش ذوالانواری و همکاران [۱] در ۳۰ ویژگی انتخابی توسط PCA در مجموعه داده‌ی UNSW-NB15 توسط پژوهش ما وجود دارد، بنابراین در روش ما نیز سیستم تشخیص نفوذ قابل اتکایی در حوزه‌ی اینترنت‌اشیاء صنعتی خواهیم داشت.

۲,۴ تحلیل‌گر مبتنی‌بر شبکه عصبی CNN-LSTM

در این مرحله مهم‌ترین مؤلفه این معماری قرار دارد که بخش موتور تحلیل‌گر آن است. این مرحله وظیفه پردازش مجموعه داده و در نهایت دسته‌بندی آن‌ها را برعهده دارد. این مرحله به چند گام تقسیم می‌شود:

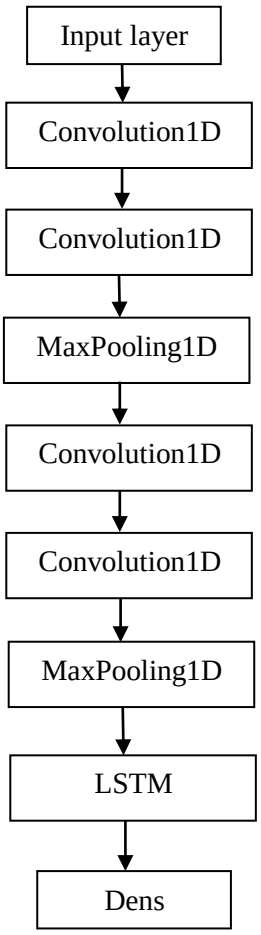
- بهینه‌سازی فرا پارامترهای شبکه عصبیِ CNN-LSTM با استفاده از الگوریتم PSO-WOA
  - نحوه‌ی ترکیب و تعویضِ الگوریتم‌های بهینه‌سازی PSO و WOA
- ساخت مدل مبتنی بر شبکه عصبی

۱,۲,۴ بهینه‌سازی فرا پارامترهای CNN-LSTM با استفاده از الگوریتم PSO-WOA

یکی از چالش‌های مهم در شبکه‌های عصبی تخمین فرا پارامترهای بهینه است، که شبکه‌ی عصبی CNN-LSTM نیز از این قاعده مستثنی نمی‌باشد. شبکه عصبی CNN-LSTM از چندین روال تشکیل شده است. بعد از دریافت ورودی، یکی از روال‌ها تعیین ساختار شبکه عصبی است، که تعیین فرا پارامترهای شبکه نیز جزء این روال است، با توجه معماری‌های مختلف ارائه شده روی ترکیب

[1] nominal
[2] interger
[3] timestamp
[4] binary
[5] Label Encoding
[6] variance
[7] Z-score

چندین شبکه‌ی عصبی CNN [۲۷] ، معماری GoogleNet دارای کارایی مناسبی در زمینه دقت، زمان، محدودیت انرژی بود و ما با الگو گرفتن از یک بلوک ساختاری GoogleNet و بررسی حالت‌های مختلف برای لایه‌های شبکه عصبی، ساختاری مطابق شکل ۴ ارائه می‌دهیم که شامل چهار لایه CNN ، دو لایه MaxPooling ، یک لایه LSTM و در نهایت یک لایه فشرده‌سازی برای دسته‌بندی می‌باشد. همچنین با مشاهده‌ی کارایی بهتر شبکه عصبی در دو برابر کردن تعداد نورون‌های دو لایه‌ی آخر CNN نسبت به نورون‌های اولیه، این تغییر را نیز در ساختار اولیه لحاظ کردیم.



شکل ۴ ساختار شبکه عصبی CNN-LSTM

مسئله‌ی پیدا کردن فراپارامترهای بهینه برای شبکه عصبی CNN-LSTM به عنوان یک چالش مطرح می‌شود، لذا تعیین این فرا پارامترها در عملکرد شبکه نقش بسزایی ایفا می‌کند. چون هیچ دیدگاه و قانون خاصی وجود ندارد، معمولا تعیین فرا پارامترها به صورت آزمون و خطا تعیین شده و شبکه را ایجاد می‌کند و دفعات زیادی این عمل انجام می‌شود که تعداد این تکرارها نیز توسط کاربر مشخص می‌شود. عیب این روش این است که اولا در هر بار تکرار باید کل فضای فرضیه را جستجو کنیم تا فرا پارامترهایی انتخاب کنیم، ثانیا در هر بار تکرار باید کل مجموعه داده را برای آموزش شبکه اعمال کنیم. اما تعیین این فرا پارامترها را می‌توان به صورت خودکار توسط یک الگوریتم جستجوی اکتشافی انجام داد، که ما در این پایان‌نامه ترکیب الگوریتم‌های بهینه سازی ازدحام ذرات و وال (PSO-WOA) را پیشنهاد می‌کنیم، که هر دو الگوریتم و در نتیجه ترکیب آن‌ها، الگوریتم‌های کارا با سرعت و دقت بالا می‌باشند. هر دو الگوریتم در هر بار تکرار کل فضای فرضیه را جستجو نمی‌کنند و براساس تجربه‌های گذشته به دنبال راه‌حل‌های بهینه می‌گردند، در واقع از هوش جمعی استفاده می‌کنند. هوش جمعی خاصیتی سیستماتیک است که عامل‌ها به‌طور محلی باهم همکاری می‌نمایند و رفتار جمعی تمام عامل‌ها باعث یک همگرایی در نقطه‌ای نزدیک به جواب بهینه سراسری می‌شود. نقطه قوت این الگوریتم‌ها عدم نیاز به یک کنترل سراسری می‌باشد. الگوریتم PSO مبتنی بر حرکت و سرعت جابجایی ذرات و نیز الگوریتم WOA مبتنی بر مکانیزم کوچک کردن محاصره(فضای مسئله) و بروز رسانی موقعیت مارپیچی عمل می‌کند، هر دو الگوریتم پیچیدگی زمانی و سرعت بالایی دارند. هر دو الگوریتم با در نظر داشتن تمام ابعاد مسئله ابتدا کل فضای مسئله را جستجو می‌کنند و به‌مرور زمان فضای جستجوی خود را به سمت بهترین مکان‌هایی که مشاهده کرده محدود می‌کنند. این ویژگی منحصر به‌فرد باعث قوت و کارایی بالای الگوریتم در عین حفظ کارایی و دقت در به دست آوردن فرا پارامترهای بهینه‌ی شبکهِ عصبی می‌باشد [۲۲] [۲۱]. در واقع هدف ما این است که در زمان کم و با دقت بیشتری روال کار سیستم را با دو الگوریتم بهینه کنیم. برای تحقق این امر ما مجموعه‌ای از متغیرهای تصادفی را در طول زمان مشخص می‌کنیم و آن‌ها را به‌عنوان فرا پارامترهای شبکه در نظر می‌گیریم. برای تعیین فرا پارامترها در هر شبکه‌ی عصبی، تعیین یکسری فرا پارامترها مانند تعداد نورون‌ها در هر لایه، تعداد لایه‌ها، انتخاب الگوریتم گرادیان کاهشی[1] و تعیین نرخ یادگیری[2] این الگوریتم، نرخ dropout(که برای کنار گذاشتن تعدادی نورون در هر مرحله برای جلوگیری از بیش‌برازش[3] در شبکه عصبی استفاده می‌شود)، مقدار لایه‌ی ادغام[4] (باعث کاهش ویژگی‌ها، تعداد پارامترهای یادگیری و میزان محاسبات شبکه می‌شود) و بایاس[5] (مقدار وزن‌دهی اولیه نورون‌ها) در شبکه‌ی عصبی بانظارت در عملکرد شبکه عصبی بسیار حائز اهمیت است، لذا در شبکه عصبی CNN-LSTM نیز این فرا پارامترها مهم

[1] Gradient descent
[2] Learning rate
[3] overfitting
[4] Pooling layer
[5] bias

هستند. درصورتی که همه‌ی فرا پارامترهای ذکر شده برای بهینه‌سازی انتخاب شوند، فضای حل مسئله بیش ازحد پیچیده خواهد شد. بنابراین در این پژوهش مطابق جدول ۲ برخی از مهم‌ترین فرا پارامترها، یعنی تعداد نورون‌ها در هر لایه، نرخ یادگیری، نرخ dropout، مقدار لایه ادغام و بایاس را به عنوان فرا پارامترهای انتخاب شده برای بهینه‌سازی درنظر می‌گیریم. در ادامه به تابع هزینه روش PWCL، نحوه‌ی بهینه‌سازی فرا پارامترهای شبکه عصبی توسط الگوریتم پیشنهادی به نام PSO-WOA و سپس به نحوه تعویض دو الگوریتم می‌پردازیم.

جدول ۲ فرا پارامترهای انتخابی برای بهینه‌سازی در شبکه عصبی

شماره	نام فرا پارامتر	توصیف
۱	Neurons	تعداد نورون‌های اولیه‌ی در نظر گرفته شده برای شبکه عصبی
۲	Learning rate	ضریب گام‌هایی که به سمت حل مسئله برداشته می‌شود
۳	Dropout	کنار گذاشتن تعدادی نورون در هر مرحله
۴	Poolsize	مقدار لایه ادغام
۵	Bias	مقدار وزن‌دهی اولیه نورون‌ها

در شبکه‌های کنترل صنعتی به علت نیاز به ارتباط بلادرنگ، زمان بسیار اهمیت پیدا می‌کند و تاخیر باتوجه به ماهیت این سیستم‌ها خسارت جبران ناپذیری به وجود می‌آورد؛ همچنین در سیستم‌های کنترل صنعتی اگر حمله‌ای به صورت کشف نشده باقی بماند به معنای اعمال مخرب علیه این سیستم‌ها است که می‌تواند منجربه نتایج فاجعه‌باری شود. برای محاسبه‌ی تابع هزینه، باتوجه به مطالب بیان شده و با توجه به پیاده‌سازی سیستم تشخیص نفوذ در سیستم‌های کنترل صنعتیِ مبتنی بر اینترنت‌اشیاء، افزایش نرخ تشخیص[1] و کاهش نرخ منفی کاذب[2]، زمان و همچنین نرخ درستی[3] (به‌دلیل اینکه میزان مفید بودن الگوریتم‌ها را نشان می‌دهد) برای ما اهمیت دارد. باتوجه به مطالب بیان شده، تابع هزینه‌ی مورد نظر مطابق با فرمول (۵) محاسبه می‌شود. تابع هزینه برابر با مجموع هزینه‌های زمان آموزش، نرخ تشخیص و نرخ درستی می‌باشد. هزینه‌ی زمان آموزش $(Cost_{Time\_train})$ مطابق فرمول (۲) برای هر دقیقه طول کشیدن آموزش یک واحد افزایش می‌یابد. هزینه‌ی نرخ تشخیص $(Cost_{Detection\_rate})$ مطابق فرمول (۳) برابر با مقدار خطای تشخیص و هزینه‌ی درستی $(Cost_{Accuracy\_rate})$ مطابق فرمول (۴) برابر با مقدار خطای درستی می‌باشد. در نهایت مجموعه هزینه‌ی فرمول‌های (۲)، (۳) و (۴)، هزینه‌ی کلی مدنظر ما یعنی فرمول (۵) را تشکیل می‌دهند. بنابراین تابع هزینه مورد نظر که الگوریتم فرابتکاری PSO-WOA درپی کاهش آن است، مطابق فرمول (۵) محاسبه می‌کنیم.

$$Cost_{Time\_Train} = Time_{Train}(seconds) / 60 \quad (۲)$$

$$Cost_{Detection\_rate} = 100 - Detection\_rate \quad (۳)$$

$$Cost_{Accuracy\_rate} = 100 - Accuracy \quad (۴)$$

$$Cost = Cost_{Time\_Train} + Cost_{Detection\_rate} + Cost_{Accuracy\_rate} \quad (۵)$$

در روش PWCL بخش موتور تحلیل‌گر مبتنی بر شبکه عصبی بدین صورت عمل می‌کند که در مرحله اول با استفاده از الگوریتم PSO-WOA مقادیری به فرا پارامترهای شبکه عصبی CNN-LSTM می‌دهد. در مرحله دوم شبکه عصبی اجرا شده و تابع هزینه محاسبه می‌شود. در مرحله سوم مقدار هزینه‌ی محاسبه شده به الگوریتم PSO-WOA برگشت داده می‌شود. در مرحله چهارم با توجه به مقدار هزینه، مجدد الگوریتم PSO-WOA فرا پارامترها را مقداردهی کرده و درپی کاهش تابع هزینه می‌باشد. درنهایت، با رسیدن به حدآستانه یا حداکثر تعداد تکرار فرا پارامترهای بهینه تحویل مرحله‌ی بعد می‌شوند. مقداردهی فرا پارامترها با استفاده از ترکیب دو الگوریتم بهینه‌سازی PSO و WOA انجام می‌شود، که در ادامه در مورد نحوه ترکیب و تعویض دو الگوریتم PSO و WOA و نحوه تعیین حدآستانه و حداکثر تعداد تکرار توضیح داده می‌شود.

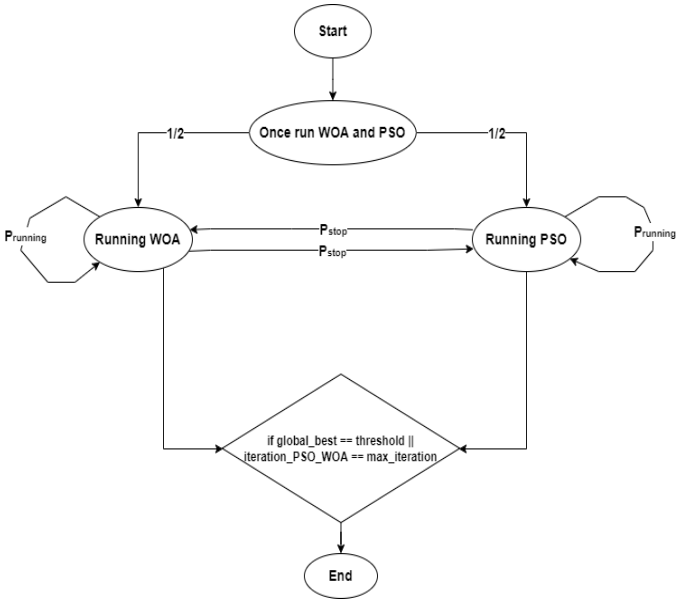
۱،۱،۲،۴ نحوه‌ی ترکیب و تعویضِ الگوریتم‌های بهینه‌سازی PSO و WOA

مطابق شکل ۵ برای ترکیب و تعویض الگوریتم‌های بهینه‌سازی، با توجه به این‌که وضعیت آینده الگوریتم فقط به وضعیت فعلیِ الگوریتم بستگی دارد، از زنجیره مارکوف ایده می‌گیریم.

[1] Detection Rate

[2] False Negative rate

[3] Accuracy



شکل ۵ نحوه ترکیب و تعویض الگوریتم PSO و WOA

برای ترکیب الگوریتم‌های بهینه‌سازی PSO و WOA در ابتدا هر الگوریتم یک مرتبه اجرا شده و سپس با احتمال برابر یکی از الگوریتم‌های PSO یا WOA اجرا می‌شود. در هر تکرار با احتمال $P_{running}$ الگوریتم در حال اجرا به اجرای خودش ادامه می‌دهد و با احتمال P_{stop} به سمت الگوریتم متوقف رفته و تعویض الگوریتم اتفاق می‌افتد. برای محاسبه‌ی احتمال $P_{running}$ و P_{stop} سه پارامتر تعداد تکرار هر الگوریتم، بهبود تابع هزینه‌ی الگوریتم درحال اجرا با مقدار قبلی و مقایسه‌ی تابع هزینه‌ی دو الگوریتم را در نظر می‌گیریم. برای محاسبه‌ی بردار احتمالِ تعداد تکرار الگوریتم مطابق جدول ۳ و معادله‌های شماره‌ی (۱) و (۲) عمل می‌کنیم. در معادله (۱) $iteration_r$ و $iteration_s$ به ترتیب تعداد تکرار الگوریتم در حال اجرا و متوقف بوده و $\max\_iteration$ حداکثر تعداد تکرار در نظر گرفته شده برای الگوریتم PSO-WOA می‌باشد. در معادله (۲) P_{ir} و P_{is} به ترتیب احتمالِ تعداد تکرار برای الگوریتم در حال اجرا و متوقف می‌باشد، که در ابتدا احتمال هر دو برابر درنظر گرفته شده و سپس باتوجه به مقدار $difference_{iter}$ محاسبه می‌شود.

برای محاسبه‌ی بردار احتمالِ بهبود تابع هزینه‌ی الگوریتم در حال اجرا با مقدار قبلی مطابق جدول ۳ و معادله‌ی شماره (۳) عمل می‌کنیم. P_{icr} و P_{ics} به ترتیب بیانگر احتمال بهبود تابع هزینه برای الگوریتم در حال اجرا و متوقف می‌باشد. شمارنده‌ی $counter_{ir}$ تعداد تکرار الگوریتم در حال اجرا در وضعیت فعلی و شمارنده‌ی $counter_{ics}$ در هر مرحله در صورتی که شرط $cost_r(t) \geq cost_r(t-1)$ برقرار باشد به اضافه‌ی یک می‌شود، این دو شمارنده بعد از تعویض الگوریتم صفر می‌شوند. $cost_r$ نیز تابع هزینه‌ی محاسبه شده برای الگوریتم در حال اجرا می‌باشد.

برای محاسبه‌ی بردار احتمالِ مقایسه‌ی تابع هزینه‌ی دو الگوریتم مطابق جدول ۳ معادله‌ی شماره (۴) عمل می‌کنیم. P_{ccr} و P_{ccs} به ترتیب احتمالِ مقایسه‌ی تابع هزینه برای الگوریتم در حال اجرا و متوقف می‌باشد. شمارنده‌ی $counter_{ccs}$ در هر مرحله در صورتی که شرط $cost_r \geq cost_s$ برقرار باشد به اضافه‌ی یک شده و این شمارنده بعد از تعویض الگوریتم صفر می‌شود. $cost_s$ نیز تابع هزینه محاسبه شده برای الگوریتم متوقف می‌باشد.

با توجه به مطالب گفته شده، سه بردار احتمال مورد نظر ماتریس استوکستیکی مطابق جدول ۳ و معادله‌ی شماره (۵) تشکیل می‌دهند.

با توجه به آزمایشات انجام شده برای پارامترهای مورد نظر بردار احتمالی را مطابق جدول ۳ و معادله‌ی شماره (۶) در نظر می‌گیریم. P_i ، P_{ic} و P_{cc} به ترتیب احتمال تعداد تکرار هر الگوریتم، بهبود تابع هزینه‌ی الگوریتم با مقدار قبلی و مقایسه‌ی تابع هزینه‌ی دو الگوریتم می‌باشد. احتمال P_i به این دلیل بالاتر است که در آزمایشات انجام شده به طور معمول تابع هزینه بعد از چندین تکرار بهبود چشمگیری را در تکرار بعدی داشت، بنابراین احتمال تعداد تکرار را بیشتر درنظر گرفتیم.

با توجه به این‌که بردار احتمال (P_{ir}, P_{is}) به تعداد تکرار الگوریتم‌ها، بردار احتمال (P_{icr}, P_{ics}) به تابع هزینه‌ی وضعیت فعلی و قبلی در الگوریتم در حال اجرا و نیز بردار احتمال (P_{ccr}, P_{ccs}) به تابع هزینه‌ی الگوریتم در حال اجرا و متوقف وابسته هستند، همچنین بردار احتمال (P_i, P_{ic}, P_{cc}) ثابت است؛ بنابراین بردارهای احتمال معرفی شده کاملا مستقل از هم‌دیگر می‌باشند. در نهایت، احتمال وقوع همزمان $P_{running}$ و P_{stop} با توجه به معادله‌های (۵) و (۶)، مطابق جدول ۳ معادله‌ی شماره (۷) محاسبه می‌شود.

جدول ۳ معادلات

شماره	معادله
(۱)	$difference_{iter} = \frac{(iteration_r - iteration_s)}{\max\_iteration}$
(۲)	$(P_{ir}, P_{is}) = \left(\frac{۱}{۲} - difference_{iter}, \frac{۱}{۲} + difference_{iter}\right)$

$(P_{icr}, P_{ics}) = \begin{cases} \left(1 - \frac{counter_{ir}}{\max\_iteration}, \frac{counter_{ir}}{\max\_iteration}\right) & if\ cost_r(t) < cost_r(t-1) \\ \left(\frac{counter_{ics}}{\max\_iteration}, 1 - \frac{counter_{ics}}{\max\_iteration}\right) & if\ cost_r(t) \geq cost_r(t-1) \end{cases}$	(۳)
$(P_{ccr}, P_{ccs}) = \begin{cases} \left(1 - \frac{counter_{ir}}{\max\_iteration}, \frac{counter_{ir}}{\max\_iteration}\right) & if\ cost_r < cost_s \\ \left(\frac{counter_{ccs}}{\max\_iteration}, 1 - \frac{counter_{ccs}}{\max\_iteration}\right) & if\ cost_r \geq cost_s \end{cases}$	(۴)
$\begin{bmatrix} P_{ir} & P_{is} \\ P_{icr} & P_{ics} \\ P_{ccr} & P_{ccs} \end{bmatrix}$	(۵)
$(P_i, P_{ic}, P_{cc}) = (\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$	(۶)
$(P_{running}, P_{stop}) = [1/2 \quad 1/4 \quad 1/4] \times \begin{bmatrix} P_{ir} & P_{is} \\ P_{icr} & P_{ics} \\ P_{ccr} & P_{ccs} \end{bmatrix} = \left[\frac{1}{2}P_{ir} + \frac{1}{4}P_{icr} + \frac{1}{4}P_{ccr} \quad \frac{1}{2}P_{is} + \frac{1}{4}P_{ics} + \frac{1}{4}P_{ccs}\right]$	(۷)

شرط خاتمه الگوریتم PSO-WOA وابسته به مقدار حد آستانه‌ی محاسبه شده توسط تابع هزینه و تعداد تکرار الگوریتم است که توسط مدیر مقداردهی می‌شود. اگر مقدار حد آستانه مقدار بالایی باشد، تعداد تکرار الگوریتم PSO-WOA به مراتب افزایش خواهد یافت، چرا که افزایش نرخ تشخیص و کاهش نرخ منفی کاذبِ محاسبه شده توسط تابع هزینه، ملاک خاتمه الگوریتم PSO-WOA می‌باشد. اگر هم مقدار حد آستانه، مقدار کوچکی در نظر گرفته شود، تعداد تکرار الگوریتم PSO-WOA خیلی کمتر خواهد بود و سیستم تشخیص نفوذ قابل اتکایی طبق اهداف مسئله‌ی تعیین شده نخواهیم داشت. لذا این حد آستانه نیز باید توسط تابع هزینه با مقدار قابل قبولی محاسبه شود و نحوه محاسبه این مقدار نیز به صورت آزمون و خطا طی چند مرحله توسط مدیر تعیین می‌شود. بنابراین برای تعیین مقدار حد آستانه و تعداد تکرار، الگوریتم PSO-WOA را در چند مرحله و با مقادیر واقعی مورد آزمایش و ارزیابی قرار می‌دهیم. مقدار این حد آستانه و تعداد تکرار بدین صورت تعیین می‌شود که اگر بعد از چند مرحله اجرای الگوریتم PSO-WOA بهبودی در تابع هزینه صورت نگیرد، مدیر باتوجه به مشاهدات می‌تواند مقدار مشخصی را برای تعداد تکرار یا مقدار حد آستانه درنظر بگیرد. حتی می‌توان به صورت دلخواه شرطی در الگوریتم لحاظ کرد که اگر تابع هزینه مثلا بعد از ۵ یا ۶ مرحله اجرای الگوریتم PSO-WOA بهبودی پیدا نکند، الگوریتم متوقف شود. گزارش تعیین مقدار حد آستانه و تعداد تکرار در فصل ارزیابی روش پیشنهادی نمایش داده شده است.

۲،۲،۴ ساخت مدل مبتنی بر شبکه عصبی

بعد از تعیین فرا پارامترهای شبکه عصبی CNN-LSTM روال ایجادِ ساختار شبکه عصبی تمام و وارد روال آموزش شبکه عصبی CNN-LSTM می‌شویم. در این روال براساس فرا پارامترهای تعیین شده توسط الگوریتم PSO-WOA، مجموعه داده ورودی به شبکه اعمال شده و داده‌های آموزشی را براساس برچسبی که دارند به شبکه آموزش می‌دهیم.

پس از آموزش شبکه با توجه به همان فرا پارامترهای تعیین شده، مجموعه داده‌ی آزمایشی را وارد شبکه عصبی می‌کنیم. درنهایت با استفاده از شبکه عصبی CNN-LSTM و با استفاده از فرا پارامترهای بهینه‌ی تعیین شده، به افزایش نرخ تشخیص و کاهش نرخ منفی کاذب و کاهش زمان آموزش و آزمایش دست پیدا خواهیم کرد.

۳،۴ مدیر و کنترل‌کننده

در مؤلفه سوم از معماری سیستم تشخیص نفوذ در اینترنت اشیاء صنعتی مبتنی بر یادگیری عمیق، مدیر و کنترل کننده قرار دارد. در این مؤلفه مقدار دهی یک‌سری پارامترها، مدیریت فضای کاربر و اجرای فرمان‌ها صادر شده از سوی کاربر با برقراری ارتباط با مؤلفه‌های سیستم می‌باشد. لذا مشخص نمودن مقدار خاتمه الگوریتم PSO-WOA برای تعیین فرا پارامترها توسط کاربر با توجه به اینکه بهبودی در مقدار تابع هزینه بعداز چند مرحله اجرای الگوریتم PSO-WOA انجام نگیرد، تعیین می‌شود.

در این بخش، هدف روش PWCL، کاهش پیچیدگی محاسبات سیستم، کاهش تعداد تکرار آموزش و آزمایش شبکه عصبی و افزایش سرعت و نرخ تشخیص و کاهش نرخ منفی کاذب است، که با ارائه روش PWCL از طریق شبکه عصبی CNN-LSTM و الگوریتم بهینه‌سازی PSO-WOA معرفی و تشریح شد. سیستم تشخیص نفوذ در اینترنت اشیاء صنعتیِ مبتنی بر یادگیری عمیق، یک سیستم تشخیص نفوذ مبتنی بر شبکه عصبی بانظارت است که در مرحله آموزش آن از الگوریتم PSO-WOA جهت تعیین و بهینه‌سازی فرا پارامترهای آن استفاده شد. الگوریتم PSO-WOA با توجه به سرعت بالایی که دارد، فرا پارامترهایی را مشخص می‌کند و در ارزیابی این فرا پارامترها، شبکه عصبی CNN-LSTM را اجرا کردیم و با محاسبه تابع هزینه در شبکه عصبی و در چندین

مرحله تکرار الگوریتم، فرا پارامترهای بهینه مشخص شد. با این فرا پارامترهای بهینه روشی برای آموزش رکوردهای مجموعه داده آموزشی ایجاد شد و بعد از آموزش موفق، در مرحله آزمایش با مجموعه داده آزمایشی، میزان مؤثر بودن روش PWCL را ارزیابی می‌کنیم. این سیستم با هدف ارزیابی سیستم تشخیص نفوذ در اینترنت اشیاء صنعتیِ مبتنی بر شبکه عصبی CNN-LSTM با استفاده از الگوریتم‌های فرا ابتکاریِ PSO-WOA ارائه شد و با سیستم تشخیص نفوذ مبتنی بر شبکه‌های عصبی بانظارت دیگر مقایسه انجام گرفت، که در بخش بعدی تنایج آزمایش‌ها در مراحل مختلف سیستم مورد بررسی و ارزیابی قرار می‌گیرد.

۵. ارزیابی روش پیشنهادی

در این بخش در مورد ملاحظات فنی و پیاده‌سازی، مجموعه داده، جزئیات ارزیابی طرح پیشنهادی و همچنین مقایسه با پژوهش‌های پیشین صورت می‌گیرد.

۱,۵ ملاحظات فنی و پیاده‌سازی

برای تعیین ساختار و تعیین فرا پارامترهای شبکه عصبی با استفاده از روش PWCL و ساخت مدل مبتنی بر شبکه عصبی CNN-LSTM با استفاده از فرا پارامترهای تعیین شده توسط روش PWCL، از Google Colab pro+ استفاده شده است.

برای ارزیابی سیستم‌های تشخیص نفوذ در اینترنت‌اشیاء صنعتی از مجموعه داده UNSW-NB15 استفاده می‌کنیم.

۲,۵ مجموعه داده

این مجموعه داده دارای ۹ نوع حمله شامل Fuzzers, Analysis, Backdoors, Dos, Exploits, Generic, Reconnaissance, Shellcode و Worms است و شامل ۴۹ ویژگی می‌باشد. تعداد کل رکوردها دو میلیون و پانصد و چهل هزار و پانصد چهل و چهار (۲۵۴۰۵۴۴) می‌باشد، همچنین مجموعه داده مناسب‌تری از داده‌های اصلی شامل ۱۷۵۳۴۱ رکوردِ آموزش و ۸۲۳۳۲ رکوردِ آزمایش آماده شده است که اغلب پژوهش‌ها از این مجموعه داده استفاده می‌کنند [۲۴]. در این پژوهش نیز از مجموعه داده‌های آموزشی و آزمایشی UNSW-NB15 با ۴ ویژگی کمتر نسبت به مجموعه داده اصلی، یعنی از ۴۵ ویژگی استفاده می‌کنیم.

۳,۵ جزئیات ارزیابی طرح پیشنهادی

برای ارزیابی و تحلیل روش PWCL، از معیارهایی که در همه‌ی سیستم‌های تشخیص نفوذ براساس TP,TN,FN,FP رایج است، استفاده می‌کنیم. معیارهای ذکر شده را به اختصار شرح می‌دهیم.

- (True Positive) TP : برچسب حمله دارند و حمله تشخیص داده شده‌اند.
- (False Positive) FP : برچسب نرمال دارند ولی تراکنش‌های حمله تشخیص داده شده‌اند.
- (False Negative) FN : برچسب حمله دارند ولی تراکنش‌های نرمال تشخیص داده شده‌اند.
- (True Negative) TN : برچسب نرمال دارند و تراکنش‌های نرمال تشخیص داده شده‌اند.

۴,۵ آزمایش و ارزیابی

ساختار شبکه عصبی و مقادیر پارامترها و فرا پارامترهای آن در کارایی و بهره‌وری برای حل مسائل مختلف تاثیر زیادی دارد. در روش PWCL ابتدا یکسری آزمایش‌های اولیه برای تعیین ساختار مناسب شبکه عصبی و مقادیر مناسب فرا پارامترهای اساسی آن با استفاده از الگوریتم فرا ابتکاری PSO-WOA تعیین شده و سپس آزمایش‌های اصلی جهت ارزیابی کارایی سیستم صورت گرفته است.

در پیاده‌سازی روش PWCL، پارمترهایی که در الگوریتم PSO-WOA وجود دارد در جدول ۴ قرار داده شده است. همان‌گونه که مشاهده می‌کنید تعداد ذرات وال‌های فضای جستجو برای الگوریتم ۵ در نظر گرفته شده و تعداد تکرار بدون لحاظ کردن حد آستانه، به صورت پیش‌فرض ۲۰ در نظر گرفته شده و همچنین تعداد دوره آموزشی در شبکه عصبی CNN-LSTM در هر فراخوانی صورت گرفته شده توسط الگوریتم PSO-WOA مقدار ۵ در نظر گرفته شده است.

در ادامه به مقایسه و ارزیابی روش پیشنهادی با چهار وضعیت مختلف می‌پردازیم:

۱) استفاده از انتخاب ویژگی‌های ویژه (PCA) و بدون مرحله‌ی انتخاب ویژگی‌های ویژه
۲) نتایج الگوریتم‌های PSO و WOA به صورت جداگانه
۳) تعویض الگوریتم‌های PSO و WOA با دو روش متداولِ تعویض تصادفی و تطبیقی
۴) استفاده از ۱۰ و ۳۰ ویژگی انتخابی
۵) بررسی و تحلیل نتایج

جدول ۴ پارامترهای الگوریتم WOA-PSO

متغیر	W	C_1	C_2	Particle/Whale	Max_iteration	دوره آموزشی
توصیف	وزن اینرسی	ضریب اعتماد به pbest	ضریب اعتماد به gbest	تعداد ذرات و وال‌های فضای جستجو	تعداد تکرار فضای جستجوی حالت	تعداد دوره آموزشی شبکه عصبی
مقادیر در نظر گرفته شده	۱	۲	۲	۵	۲۰	۵

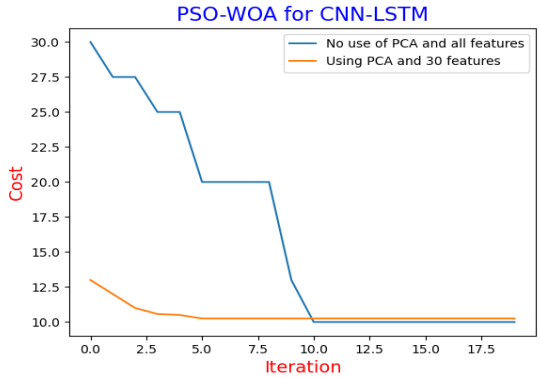
۱،۴،۵ مرحله‌ی انتخاب ویژگی‌های ویژه و بدون مرحله‌ی انتخاب ویژگی‌های ویژه

در مؤلفه‌ی پیش‌پردازنده به دو صورتِ بدون مرحله‌ی انتخاب ویژگی‌های ویژه و با مرحله‌ی انتخاب ویژگی‌های ویژه می‌توان ویژگی‌ها را به مؤلفه‌ی تحلیل‌گر مبتنی بر شبکه عصبی تحویل داد. با توجه به توضیحات فصل چهارم در این پژوهش با استفاده از روش تحلیل مؤلفه‌های اصلی (PCA) به کاهش ویژگی‌ها پرداختیم و از کل ویژگی‌ها ۳۰ ویژگی را برای آموزش و آزمایش روش PWCL انتخاب کردیم. در این قسمت به مقایسه و ارزیابی روش PWCL با استفاده از مرحله‌ی انتخاب ویژگی و بدون انتخاب ویژگی می‌پردازیم.

مطابق شکل ۶ روند بهبود تابع هزینه در روش PWCL، با استفاده از PCA و ۳۰ ویژگی انتخابی و بدون استفاده از PCA و همه ویژگی‌ها را مشاهده می‌کنید. بدون استفاده از مرحله‌ی انتخاب ویژگی‌های ویژه در تکرارهای اولیه‌ی الگوریتم PSO-WOA تابع هزینه مقدار بالاتری را دارد و در تکرار یازدهم به مقدار بهینه دست پیدا کرده و نیز زمان اجرای روش PWCL ۸۰ دقیقه طول کشیده است. در مقابل، با مرحله‌ی انتخاب ویژگی‌های ویژه در همان تکرار اولیه‌ی الگوریتم PSO-WOA تابع هزینه بهبود چشمگیری را نسبت به مرحله‌ی بدون انتخاب ویژگی‌های ویژه داشته و همچنین در تکرار ششم به مقدار بهینه دست پیدا کرده و نیز زمان اجرای روش PWCL ۶۰ دقیقه طول کشیده است. بنابراین با مرحله‌ی انتخاب ویژگی علاوه بر رسیدن تابع هزینه به مقدار مناسبی در تکرار های اولیه، پیچیدگی محاسباتی و زمان اجرا نیز بهبود پیدا کرده است.

۲،۴،۵ نتایج الگوریتم‌های PSO و WOA به صورت جداگانه

الگوریتم بهینه‌سازی PSO دارای سرعت همگرایی خوبی است ولی در فاز اکتشاف دارای محدودیت‌هایی است که می‌تواند در بهینه‌ی محلی گیر کند؛ در طرف مقابل الگوریتم WOA دارای قابلیت اکتشاف بسیار خوبی است اما در مرحله‌ی بهره‌برداری دارای محدودیت‌هایی است که منجربه سرعت همگرایی ضعیف می‌شود. در طی آزمایشات این پژوهش، متوجه قابلیت‌ها و محدودیت‌های گفته شده در الگوریتم‌های PSO و WOA شدیم. مطابق شکل ۷ نمودار تابع هزینه را در شکل‌های a ، b ، c و d با استفاده از الگوریتم‌های PSO و WOA نشان می‌دهد، همان‌طور که مشاهده می‌کنیم در شکل a الگوریتم PSO در بهینه‌ی محلی گیر کرده است ولی در شکل b الگوریتم WOA روند بهینه‌سازی را به خوبی طی کرده است. همچنین در شکل c الگوریتم PSO سرعت همگرایی خوبی داشته و در همان ابتدا توانسته به جواب بهینه

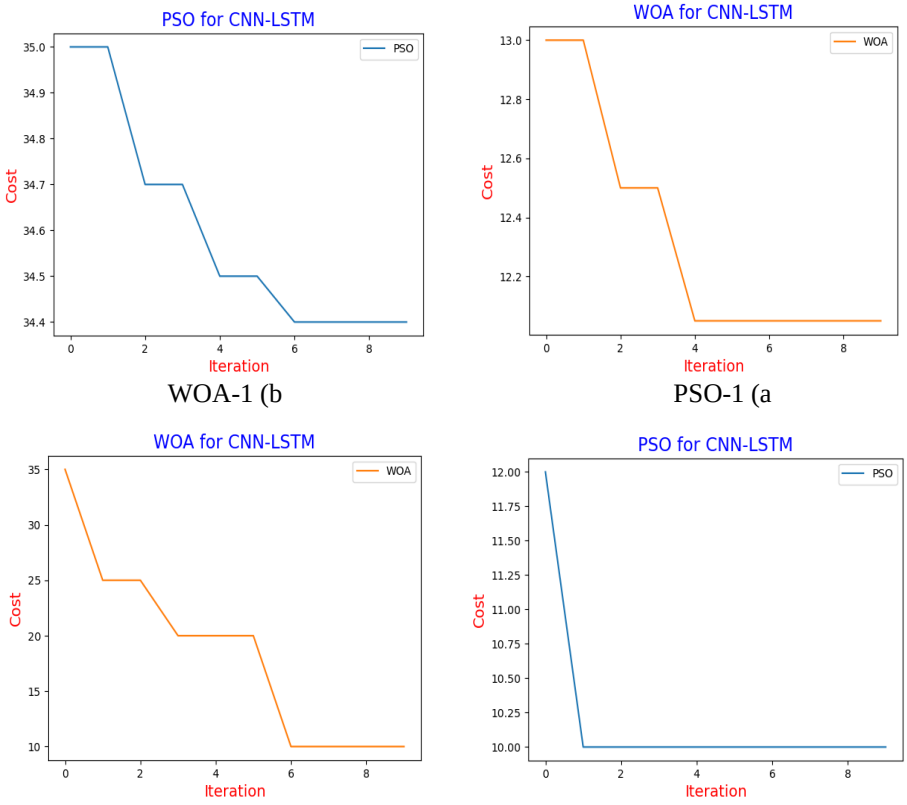


شکل ۶ روند بهبود تابع هزینه با استفاده از PCA و ۳۰ ویژگیِ انتخابی و بدون استفاده از PCA و همه ویژگی‌ها

۲،۴،۵ نتایج الگوریتم‌های PSO و WOA به صورت جداگانه

الگوریتم بهینه‌سازی PSO دارای سرعت همگرایی خوبی است ولی در فاز اکتشاف دارای محدودیت‌هایی است که می‌تواند در بهینه‌ی محلی گیر کند؛ در طرف مقابل الگوریتم WOA دارای قابلیت اکتشاف بسیار خوبی است اما در مرحله‌ی بهره‌برداری دارای محدودیت‌هایی است که منجربه سرعت همگرایی ضعیف می‌شود. در طی آزمایشات این پژوهش، متوجه قابلیت‌ها و محدودیت‌های گفته شده در الگوریتم‌های PSO و WOA شدیم. مطابق شکل ۷ نمودار تابع هزینه را در شکل‌های a ، b ، c و d با استفاده از

الگوریتم‌های PSO و WOA نشان می‌دهد، همان‌طور که مشاهده می‌کنیم در شکل a الگوریتم PSO در بهینه‌ی محلی گیر کرده است ولی در شکل b الگوریتم WOA روند بهینه‌سازی را به خوبی طی کرده است. همچنین در شکل c الگوریتم PSO سرعت همگرایی خوبی داشته و در همان ابتدا توانسته به جواب بهینه مناسبی دست پیدا کند ولی در طرف مقابل در شکل d الگوریتم WOA به دلیل سرعت همگرایی ضعیف بعد از چند مرحله به جواب بهینه دست پیدا کرده است. بنابراین استفاده‌ی ترکیبی از الگوریتم‌های بهینه‌سازی PSO و WOA می‌تواند نقاط ضعف هرکدام را پوشش داده و در زمان کمتری به جواب بهینه دست پیدا کند.



PSO-1 (a WOA-1 (b

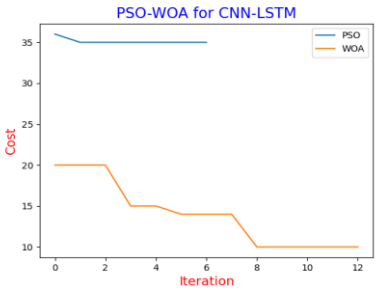
PSO-2 (c WOA-2 (d

شکل ۷ روند بهبود تابع هزینه به ترتیب از سمت راست به چپ در الگوریتم‌های PSO و WOA

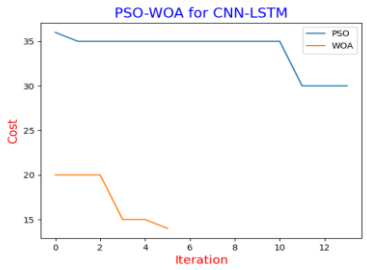
۳,۴,۵ تعویض الگوریتم‌های PSO و WOA با دو روش متداولِ تعویض تصادفی و تطبیقی

برای تعویض الگوریتم‌ها در تکرارهای مختلف دو روش متداولِ تعویض تصادفی و تطبیقی استفاده می‌شود. در روش تصادفی در هر تکرار از الگوریتم‌های بهینه‌سازی، یکی به صورت تصادفی با احتمال برابر انتخاب و اجرا می‌شود. در روش تطبیقی با توجه به تعداد راه‌حل‌های بهبود یافته به ترکیب و تعویض الگوریتم پرداخته می‌شود. در این پژوهش با ایده گرفتن و از این روش و با توجه به توضیحات فصل چهارم و آزمایشات انجام شده، در هر تکرار برای هر الگوریتم احتمالی را باتوجه به پارامترهای تعداد تکرار و تابع هزینه اختصاص می‌دهد و براساس احتمالِ محاسبه شده تکرار بعدی و تعویض الگوریتم‌ها اتفاق می‌افتد. برای مشاهده‌ی کارایی روش تطبیقی، در ادامه هر دو روش را مورد ارزیابی و مقایسه قرار می‌دهیم.

شکل ۸ نمودار تابع هزینه را در شکل‌های a و b در روش‌های تطبیقی و تصادفی نشان می‌دهد، همان‌طور که مشاهده می‌کنیم در شکل a الگوریتم PSO احتمالا در یک بهینه‌ی محلی گیر کرده و در مقابل در شکل b الگوریتم WOA روند بهبود تابع هزینه را به خوبی طی می‌کند. در روش تصادفی الگوریتم WOA با توجه به این‌که روند بهتری را در کمینه‌سازی تابع هزینه دارد ولی با توجه به تصادفی انتخاب و اجرا شدن، کمتر اجرا شده و تابع هزینه به خوبی کمینه نشده است. در طرف مقابل، روش تطبیقی روند تکرار بعدی و تعویض را بهتر انجام داده و کمینه‌سازی تابع هزینه بهتر صورت گرفته است. بنابراین در این پژوهش از روش تطبیقی با توجه به توضیحاتی که در مورد نحوه ترکیب و تعویضِ الگوریتم‌ها در فصل چهارم داده شده، استفاده می‌کنیم.



a) تطبیقی



b) تصادفی

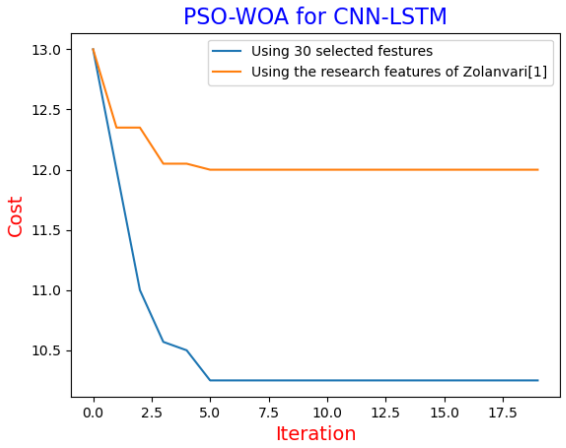
شکل ۸ روند بهبود تابع هزینه به ترتیب از سمت راست به چپ در روش‌های تطبیقی و تصادفی

۴،۴،۵ استفاده از ۱۰ و ۳۰ ویژگی انتخابی

یک گام مهم در آموزش، انتخاب و استخراج ویژگی از ترافیک در شبکه‌های مورد بررسی می‌باشد. در پژوهش ذوالانواری و همکاران [۱] با ایجاد یک بستر تست واقعی در شبکه‌ی اینترنت‌اشیاء صنعتی و همچنین پیاده‌سازی سناریوهای واقعیِ اینترنت‌اشیاء صنعتی، یک سیستم تشخیص نفوذ قابل اتکایی را ارائه داده‌اند. به‌وسیله‌ی انتخاب و استخراج ویژگی‌های بالقوه، اهمیت ویژگی‌های مختلف در تشخیص ترافیک عادی از ترافیک حمله در شبکه اینترنت اشیاء صنعتی بررسی شده و در نهایت، ۲۳ ویژگی مهم در نظر گرفته شده است. اکثر این ۲۳ ویژگی در ۳۰ ویژگی انتخابی از مجموعه داده‌ی UNSW-NB15 وجود دارد. برای مقایسه و ارزیابی جمع برخی از ویژگی‌ها از پژوهش ذوالانواری و همکاران [۱] در مجموعه داده‌ی UNSW-NB15 وجود نداشت، که محاسبه شده و به مجموعه‌ی ویژگی‌ها اضافه شده است. به عنوان مثال ویژگی‌های Sloss و Dloss نشان دهنده‌ی بسته‌های مجددا ارسال شده یا حذف شده می‌باشد. با توجه به وجود نداشتن ویژگی Tloss در مجموعه داده‌ی UNSW-NB15، که نشان‌دهنده‌ی جمع ویژگی‌های Sloss و Dloss است، این ویژگی محاسبه شده و به مجموعه‌ی ویژگی‌ها برای ارزیابی و مقایسه اضافه شده است.

مجموعه ویژگی‌های پژوهش ذوالانواری و همکاران [۱] و ویژگی‌ها در نظر گرفته شده در پیوست قابل مشاهده است. با اجرای PCA روی مجموعه ویژگی‌های پژوهش [۱] به این نتیجه می‌رسیم که با ۱۰ ویژگی حدود ۹۹ درصد از اطلاعات موجود در کل ویژگی‌ها قابل ارائه می‌باشد. بنابراین در ادامه به بررسی تابع هزینه‌ی ۱۰ و ۳۰ ویژگی می‌پردازیم.

مطابق شکل ۹ روند بهبود تابع هزینه را با استفاده از ۳۰ ویژگی و ۱۰ ویژگی در نظر گرفته شده نشان می‌دهد. با استفاده از ۳۰ ویژگی تابع هزینه بهبود تقریبا بهتری داشته است ولی در مقابل زمان اجرای روش PWCL با استفاده از ۳۰ ویژگی ۶۰ دقیقه طول کشیده ولی با استفاده از ۱۰ ویژگی ۴۵ دقیقه طول کشیده است. در هر دو روش فرا پارامترهای شبکه عصبی در حدود ۶ تکرار تعیین شده است، که با ۳۰ ویژگی حدود ۲۰ دقیقه و با ۱۰ ویژگی حدود ۱۵ دقیقه طول کشیده است. به دلیل اینکه به تابع هزینه‌ی بهتری با ۳۰ ویژگی انتخابی دست یافته‌ایم، در ادامه از ۳۰ ویژگی برای آموزش و آزمایش روش پیشنهادی استفاده کرده‌ایم.



شکل ۹ روند بهبود تابع هزینه با استفاده از ۳۰ ویژگی و ۱۰ ویژگی انتخابی با استفاده از پژوهش [۱]

با توجه به توضیحاتی که در مورد وضعیت‌های مختلف روش پیشنهادی ارائه شد، در ادامه به مقایسه و ارزیابی نتایج به دست آمده می‌پردازیم.

فرا پارامترهای بهینه‌شده برای شبکه عصبی CNN-LSTM با استفاده از الگوریتم PSO-WOA در جدول ۵ نمایش داده شده است.

جدول ۵ فرا پارامترهای بهینه شده برای شبکه عصبی CNN-LSTM با استفاده از الگوریتم WOA-PSO

نام فرا پارامتر	توصیف	مقادیر در نظر گرفته شده در شبکه عصبی CNN-LSTM
LSTM_Neurons	نورون‌های اولیه‌ی در نظر گرفته شده برای شبکه‌ی عصبی LSTM	۱۳

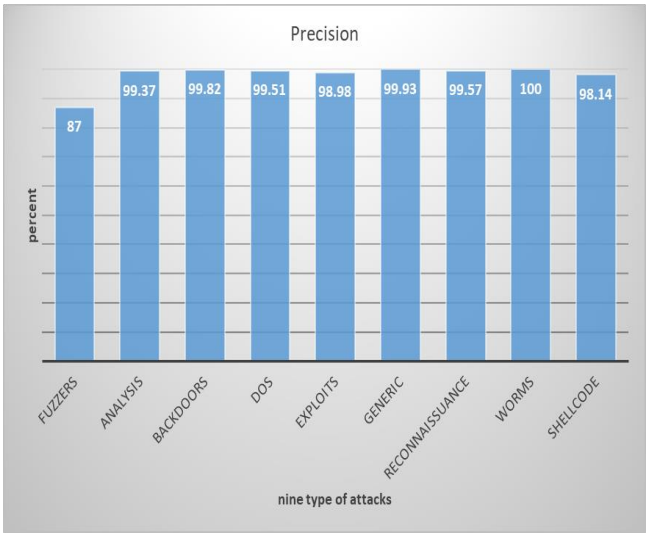
CNN_ Neurons	نورون‌های اولیه‌ی در نظر گرفته شده برای شبکه‌ی عصبی CNN	۱۱
Dropout	کنار گذاشتن تعدادی نورون در هر مرحله	۰,۴۲۳۵۵
Learning rate	ضریب گام‌هایی که به سمت حل مسئله برداشته می‌شود	۰,۰۰۳۸۵۶
Poolsize	مقدار لایه ادغام	۳
bias	مقدار وزن‌دهی اولیه نورون‌ها	۰,۰۸۳۶۰

جدول ۶ نتایج معیارهای ارزیابی

مجموعه داده	معیار ارزیابی	درصد موفقیت در زمان آموزش	درصد موفقیت در زمان آزمایش
UNSW-NB15	Detection rate	٪۹۸,۵۴	٪۹۸
	Accuracy	٪۹۴	٪۸۳,۹۰
	$Precision_p$	٪۹۳,۷۴	٪۷۷,۸۳
	$recall_p$	٪۹۸,۵	٪۹۸
	$F1-score_p$	٪۹۵,۶۹	٪۸۷,۱۲
	Error_rate	٪۶	٪۱۶,۱
	FNR	٪۱,۴۶	٪۲

مطابق جدول ۶ نتایج معیارهای ارزیابی در مدل شبکه عصبی CNN-LSTM از مجموعه داده UNSW-NB15 با ده دوره آموزشی[1] با زمان آموزش ۷۵ ثانیه و زمان آزمایش ۱۵ ثانیه نشان داده شده است.

شکل ۱۰ دقت تشخیص برای انواع مختلف حملات در مجموعه داده UNSW-NB15 را نشان می‌دهد، که به‌طور کلی روش PWCL تشخیص قابل قبولی در تشخیص همه‌ی نه نوع حمله ارائه کرده است.



شکل ۱۰ دقت تشخیص نه نوع حمله با ۳۰ ویژگی از مجموعه داده UNSW-NB15

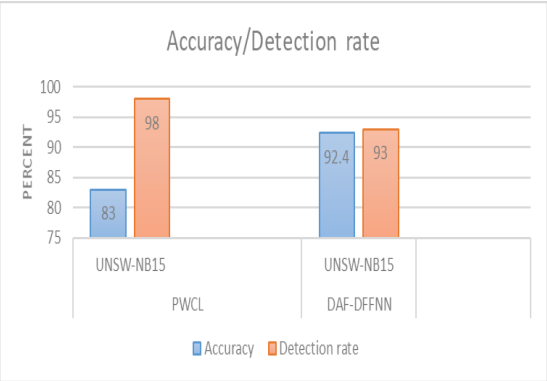
۵,۴,۵ بررسی و تحلیل نتایج

در این پژوهش سیستم تشخیص نفوذ مبتنی بر اینترنت‌اشیاء صنعتی مطابق با پژوهش DAF-DFFNN [۲] پیاده‌سازی شده و شبکه عصبی مورد نظر با توجه به مجموع مقالات ارائه شده در پیشینه تحقیق انتخاب شده است و همچنین نتایج به صورت دقیق با دو مقاله‌ی DAF-DFFNN [۲] و Intelligent-IDS [۱۶] مقایسه کرده و به صورت کلی با بقیه‌ی مقالات ارائه شده در پیشینه تحقیق بررسی می‌کنیم.

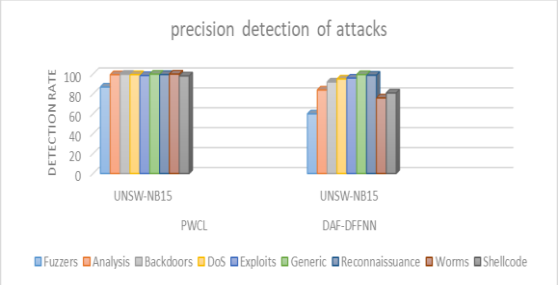
مطابق با شکل ۱۱ روش PWCL در مقایسه با DAF-DFFNN [۲] در مجموعه داده‌ی UNSW-NB15 دارای نرخ تشخیص بالاتر و دقت پایین‌تری است در شکل ۱۲ دقت تشخیص انواع حملات را در هر دو پژوهش با مجموعه داده UNSW-NB15 مشاهده می‌کنیم که روش PWCL عملکرد به مراتب بهتری داشته است.

سه نکته حائز اهمیت است: الف) در روش PWCL در تعداد دور آموزشیِ ۱۰ دوره آموزشی و در پژوهش DAF-DFFNN [۲] تعداد ۱۰۰۰ دور آموزشی مدل آموزش دیده است و به این نتایج دست پیدا کرده‌اند. پ) در پژوهش DAF-DFFNN [۲] از کل ویژگی‌ها برای آموزش و آزمایش استفاده کرده است ولی روش PWCL با استفاده از PCA و انتخاب ویژگی‌های ویژه، $\frac{1}{3}$ از تمام ویژگی‌ها کم کرده‌ایم. پ) در پژوهش DAF-DFFNN [۲] از کل مجموعه داده برای آموزش و آزمایش استفاده شده ولی در مدل PWCL از داده‌ی آموزشی و آزمایشی مخصوصِ هر مجموعه داده استفاده کرده‌ایم که با این کار نتایج قابل قبول‌تری نسبت به روش DAF-DFFNN [۲] خواهیم داشت.

[1] epochs

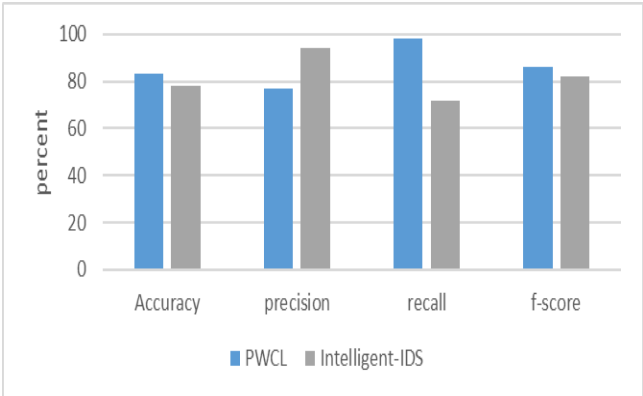


شکل ۱۱ ارزیابی روش PWCL با پژوهش DAF-DFFNN [۲]



شکل ۱۲ دقت تشخیص انواع حملات در روش PWCL و پژوهش DAF-DFFNN [۲] با مجموعه داده UNSW-NB15

مطابق شکل ۱۳ روش PWCL در مقایسه با Intelligent-IDS [۱۶] در مجموعه داده UNSW-NB15 دارای درستی، فراخوانی و f-score بالاتر و دقت پایین‌تری است. در پژوهش PWCL به معیارهای ارزیابی بهتری نسبت به پژوهش Intelligent-IDS [۱۶] دست پیدا کرده‌ایم. همچنین مزیت روش PWCL نسبت به Intelligent-IDS [۱۶] تعداد دوره آموزشی و تعیین فراپارامترهای شبکه عصبی است، که روش PWCL در ۱۰ دوره آموزشی آموزش دیده و فراپارامترهای شبکه عصبی به‌صورت خودکار تعیین شده ولی پژوهش Intelligent-IDS [۱۶] در ۱۰۰ دوره آموزشی آموزش دیده و فراپارامترهای شبکه عصبی به صورت دستی تعیین شده است.



شکل ۱۳ ارزیابی روش PWCL با پژوهش intelligent-IDS [۱۶]

در مورد سایر پژوهش‌های ذکر شده در پیشینه تحقیق می‌توان بیان کرد که روش PWCL در مقایسه با DBN-PNN [۱۴] توانسته است با استفاده از ترکیب الگوریتم‌های فراابتکاری PSO و WOA محدودیت‌های الگوریتم فراابتکاری PSO را برطرف کرده و سیستم تشخیص نفوذِ جامع و قابل قبولی ارائه دهد. همچنین با توجه به مقایسه‌ی صورت گرفته در جدول ۱، مهم‌ترین مزیت روش PWCL نسبت به پژوهش‌های TR-IDS [۱۲]، C-LSTM [۱۸]، گونزالو دلاتوره پارا و همکاران [۱۳]، هوان‌یانگ و همکارانش [۱۵]، RNN-IDS [۱۹]، SDN-DNN [۲۰] و HDRaNN [۱۷] خودکارسازیِ تنظیم فرا پارامترهای شبکه عصبی می‌باشد.

۶. خلاصه و نتیجه‌گیری

در این پژوهش یک سیستم تشخیص نفوذ مبتنی بر اینترنت‌اشیاء صنعتی ارائه شد، که این سیستم براساس شبکه‌های عصبی بهینه شده است. بهینه‌سازی فرا پارامترهای شبکه عصبی براساس الگوریتم‌های فرا ابتکاری توده ذرات و وال انجام شد و از این طریق، انتخاب فرا پارامترهای شبکه عصبی برای آموزش و آزمایش با روش پیشنهادی صورت پذیرفت. این روش انعطاف‌پذیری بالایی برای تغییرات توسط مدیریت سیستم به‌وسیله تابع هزینه مورد نظر به ما می‌دهد و همچنین نرخ تشخیص و نرخ منفی کاذب نیز بهبود پیدا کرده است. همچنین قابلیت تطبیق و پیاده‌سازی سیستم تشخیص نفوذِ مدنظر در سایر شبکه‌ها نیز وجود دارد. نتایج نشان می‌دهد که روش PWCL در مقایسه با روش‌های دیگر بهبودی لازم را با توجه به تعداد کم دوره‌های آموزشی برای سیستم فراهم آورده است. با توجه به افزایش و پیچیدگی حملات سایبری روی سیستم‌های کنترل صنعتیِ دارای حفظ و پردازش اطلاعات حساس و اهداف اصلی حملات در زیرساخت‌های بحرانی و ملی و همچنین افزایش رو به رشد تعداد اشیاء متصل به اینترنت در سال‌ها اخیر، وجود یک سیستم تشخیص نفوذ مبتنی بر اینترنت‌اشیاء صنعتی بیش از پیش ضروری است. به همین دلیل به منظور ادامه این پژوهش در آینده می‌توان از سایر الگوریتم‌های فرا ابتکاری، به منظور انتخاب فرا پارامترهای بهینه برای شبکه عصبی در سیستم تشخیص نفوذ مبتنی بر اینترنت‌اشیاء صنعتی استفاده کرد. همچنین می‌توان به بهینه‌سازی پارامتر در الگوریتم‌های بهینه‌سازی توده ذرات و وال و بهینه‌سازی تابع هزینه با توجه به اهداف مسئله پرداخت.

مراجع

[1] Z. Maede, M. A. Teixeira, G. Lav, K. M. Khan and J. Raj, "Machine Learning Based Network Vulnerability Analysis of Industrial Internet of Things," *IEEE Internet of Things*

"TR-IDS: Anomaly-Based Intrusion Detection through Text-Convolutional Neural Network and Random Forest," *Hindawi, Security and Communication Network*, 2018.

[13] G. D. L. T. Parra, R. Rad, K.-K. R. Choo and N. Beebe, "Detecting Internet of Things Attacks using Distributed Deep Learning," *Journal of Network and Computer Applications*, vol. 163, p. 102662, 2020.

[14] Z. Guangzhen, Z. Cuixiao and Z. Lijuan, "Intrusion Detection Using Deep Belief Network and Probabilistic Neural Network," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Guangzhou, China, 2017.

[15] H. Yang, L. Cheng and M. C. Chuah, "Deep-Learning-Based Network Intrusion Detection for SCADA Systems," in *2019 IEEE Conference on Communications and Network Security (CNS)*, Washington DC, DC, USA, USA, 2019.

[16] R. Vinayakumar, A. Mamoun, K. P. Soman, P. Prabaharan, A.-N. Ameer and V. Sitalakshmi, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525 - 41550, 2019.

[17] H. Zil E., L. Shahid, A. Jawad, L. Zeba, I. Anas, Z. Zhuo, A. Fehaid and B. Fatmah, "A Hybrid Deep Random Neural Network for Cyberattack Detection in the Industrial Internet of Things," *Journals & Magazines*, vol. 9, pp. 55595 - 55605, 2021.

[18] K. Tae-Young and C. Sung-Bae, "Web traffic anomaly detection using C-LSTM neural networks," *Expert System With Applications*, vol. 106, pp. 66-76, 2018.

[19] Y. Chuanlong, Z. Yuefei, F. Jinlong and H. Xinzheng, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954 - 21961, 2017.

[20] T. Tuan A, M. Lotfi, M. Des, Z. Syed Ali Raza and G. Mounir, "Deep learning approach for Network Intrusion Detection in Software Defined Networking," in *2016 International Conference on Wireless Networks and Mobile Communications Jouranl*, pp. 6822 - 6834, 2019.

[2] A.-H. Muna, N. Moustafa and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, pp. 1-11, 2018.

[3] B. Rafael Ramos Regis, "Anomaly detection in SCADA systems: a network based approach," Centre for Telematics and Information Technology, University of Twente, PhD Thesis, 2014.

[4] P. Dimitrios, S. Panagiotis, L. Thomas and A. G. Sarigiannidis, "A Survey on SCADA Systems; Secure Protocols, Incidents, Threats and Tactics," *IEEE Communications Surveys & Tutorials*, pp. 1942 - 1976, 2020.

[5] L. Hung-jen, R. L. Chun-Hung, L. Ying-Chih and T. Kuang-Yuan, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16-24, 2013.

[6] S. G. Farhad and G. Hojjat, "A comprehensive survey: Whale Optimization Algorithm and its applications," *Swarm and Evolutionary Computation*, vol. 48, pp. 1-24, 2019.

[7] H. Turabieh, M. Mafarja and X. Li, "Iterated feature selection algorithms with layered recurrent neural network for software fault prediction," *Expert Systems with Applications*, vol. 122, pp. 27-42, 2019.

[8] M. L. Naushad, G. Koushik, C. Indronil, C. Saurav, L. B. Krishna and K. P. Prashanta, "HWPSO: A new hybrid whale-particle swarm optimization algorithm and its application in electronic design optimization problems," *Applied Intelligence*, p. 265–291, 2019.

[9] A. Sadiqui, Computer Network Security, Britain and United States: WILEY, 2020.

[10] S. Anam, A. Haider and S. Kashif, "Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges," *IEEE Access*, vol. 4, pp. 1375 - 1384, 2016.

[11] O. Campesato, Artificial Intelligence Machine Learning And Deep Learning, David Pallai, 2020.

[12] E. Min, J. Long, Q. Liu, J. Cui and W. Chen,

(WINCOM), Fez, Morocco, 2016.

[21] K. James and E. Russell, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995.

[22] M. Seyedali and L. Andrew, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016.

[23] A.-G. Mohammad Ali, M. amr, A.-A. Abdulla khalid, D. Xiaojiang, A. lhsan and G. Mohsen, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," *IEEE Communications Serveys & Tutorials*, vol. 22, no. 3, pp. 1646 - 1685, 2020.

[24] N. Mostafa, "Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic," University of New South Wales, Canberra, Australia, 2017, 2017.

[25] F. lhab and c. Xu, Data Cleaning, Association for Computing Machinery, 2019.

[26] J. Edward, A User's Guide To Principal Components, New York: John Wiley & Sons, 2005.

[27] C. Alfredo, P. Adam and C. Eugenio, "An Analysis of Deep Neural Network Models for Practical Applications," in *Computer Vision and Pattern Recognition (cs.CV)*, 2017.

پیوست

ویژگی‌های انتخابی از ترافیک در پژوهش ذوالانواری و همکاران [1]

Features	Type	Descriptions
Mean flow (mean)	Float	The average duration of the active flows
Source Port (Sport)	Integer	Source port number
Destination Port (Dport)	Integer	Destination port number
Source Packets (Spkts)	Integer	Source/ Destination packet count
Destination Packets (Dpkts)	Integer	Destination/Source packet count
Total Packets (Tpkts)	Integer	Total transaction packet count
Source Bytes (Sbytes)	Integer	Source/ Destination bytes count
Destination Bytes (Dbytes)	Integer	Destination/Source bytes count
Total Bytes (TBytes)	Float	Total transaction bytes count
Source Load (Sload)	Float	Source bits per second
Destination Load (Dload)	Float	Destination bits per seconf
Total Load (Tload)	Float	Total bits per seconf
Source Rate (Srate)	Float	Source packets per second
Destination Rate (Drate)	Float	Destination packets per second
Total Rate (Trate)	Float	Total packets per second
Source Loss (Sloss)	Float	Source packets retransmitted/dropped
Destination Loss (Dloss)	Float	Destination packets retransmitted/dropped
Total Loss (Tloss)	Float	Total packets retransmitted/dropped
Total Percent Loss (Ploss)	Float	Percent packets retransmitted/dropped
Source Jitter (ScrJitter)	Float	Source jitter in millisecond
Destination Jitter (DrcJitter)	Float	Detination jitter in millisecond
Source Interpacket (SIntPkt)	Float	Source interpacket arrival time in millisecond
Destination Interpacket (DIntPkt)	Float	Destination interpacket arrival time in millisecond

ویژگی‌های انتخابی از مجموعه داده UNSW-NB15 با توجه به پژوهش ذوالانواری و همکاران [1]

Name	Type	Description
dur	Float	Record total duration
sbytes	Integer	Source to destination transaction bytes
dbytes	Integer	Destination to source transaction bytes
TBYTES	integer	Total transaction bytes
sloss	Integer	Source packets retransmitted or dropped
dloss	Integer	Destination packets retransmitted or dropped
TLOSS	Interger	Total packets retransmitted or dropped
PLOSS	Float	Percent packets retransmitted/dropped
Sload	Float	Source bits per second
Dload	Float	Destination bits per second
TLOAD	Float	Total bits per second
Spkts	integer	Source to destination packet count
Dpkts	integer	Destination to source packet count
TPKTS	interger	Total to source packet count

Sjit	Float	Source jitter (mSec)
Djit	Float	Destination jitter (mSec)
Sintpkt	Float	Source interpacket arrival time (mSec)
Dintpkt	Float	Destination interpacket arrival time (mSec)