Scientific Journal of Electronical & Cyber Defence
Vol. 10, No. 3, Autumn 2022, Serial No. 39

A Model for Multi-Class Intrusion Detection Using Machine Learning and Dragonfly Feature Selection

M. Niaei, J. Tanha*, G. Shahmohammadi, A. R. Pourebrahimi

* Associate Professor, Faculty of Electrical and Computer Engineering, Tabriz University, Tabriz, Iran

(Received: 08/10/2021, Accepted: 22/062022)

ABSTRACT

With the increase of the network services, the number and complexity of attacks in cyberspace has increased. This problem has made network security as one of the most important challenges in the world of information technology. Intrusion detection systems are used as a very important defense method to detect network attacks, to warn network security admins.This research has proposed a model for multi-class intrusion detection system. In this model, the dragonfly algorithm is used for feature selection and the random forest algorithm is used for classification. for data analysis KDD-99 dataset has been used. The model has been tested with different machine learning and deep learning algorithms then the best algorithm has been selected. The accuracy value in the proposed method is 0/9967 The results have been compared with the results of several other studies published in authoritative articles. This comparison shows that the proposed method has a higher accuracy than most other methods.

Keywords: Intrusion Detection, Multi-Class, Feature Selection, Dragonfly Algorithm, Random Forest.

* Corresponding Author Email: Tanha@tabrizu.ac.ir

نشریه علمی "پدافند الکترونیکی و سایبری"
سال دهم، شماره ۳، پاییز ۱۴۰۱، ص ۴۲- ۳۳

علمی – پژوهشی

مدلی برای تشخیص نفوذ چندکلاسه با استفاده از یادگیری ماشین و انتخاب ویژگی سنجاقک

محمود نیائی[۱]، جعفر تنها[۲*]، غلامرضا شاهمحمدی[۳]، علیرضا پورابراهیمی[٤]

۱- دانشجوی دکترای رشته مدیریت فناوری اطلاعات، گروه مدیریت فناوری اطلاعات، دانشکده مدیریت و اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ۲- دانشیار، دانشکده مهندسی برق و کامپیوتر، دانشگاه تبریز، تبریز، ۳- دانشیار، دانشکده مهندسی برق و کامپیوتر، دانشگاه ایوانکی، سمنان، ٤- استادیار، دانشکده مدیریت و حسابداری، واحد کرج، دانشگاه آزاد اسلامی، کرج، ایران

(دریافت: ۱٤۰۰/۰۷/۱٦، پذیرش: ۱٤۰۱/۰٤/۰۱)

چکیده

با افزایش خدمات شبکه‌ای، تعداد و پیچیدگی حملات در فضای سایبر افزایش یافته است؛ لذا سامانه‌های تشخیص نفوذ که به‌منظور کشف فعالیت‌های غیرعادی در شبکه‌ها مورداستفاده قرار می‌گیرند، در سال‌های اخیر موردتوجه بسیاری از محققین قرار گرفته است. سیستم‌های تشخیص نفوذ به‌عنوان یک روش دفاعی بسیار مهم برای تشخیص حملات شبکه‌ای، به‌منظور هشدار به مسئولین شبکه یا برنامه‌های کاربردی به کار می‌رود. در این پژوهش، یک مدل برای تشخیص نفوذ چندکلاسه پیشنهاد شده است. در روش پیشنهادی از الگوریتم سنجاقک برای انتخاب ویژگی و از الگوریتم جنگل تصادفی به‌منظور دسته‌بندی استفاده شده است. داده‌های به‌کاررفته در پژوهش، مجموعه‌داده KDD-99 بوده است. مسئله با الگوریتم‌های مختلف یادگیری ماشین و یادگیری عمیق مورد آزمون قرار گرفته و بهترین الگوریتم انتخاب شده است. معیار صحت در روش پیشنهادی مقدار ۰/۹۹۶۷ به دست آمده است. نتایج پژوهش با نتایج چندین پژوهش دیگر که توسط محققان پیشنهاد شده مورد مقایسه قرار گرفته است و این مقایسه نشان می‌دهد که روش پیشنهادی نسبت به اکثر روش‌های دیگر دارای مقدار صحت بالاتری بوده است.

کلیدواژه‌ها: تشخیص نفوذ، چند کلاسه، الگوریتم سنجاقک، جنگل تصادفی، انتخاب ویژگی

۱- مقدمه

مروری بر وقایع و حوادث سال‌های اخیر کشورها، مؤید این واقعیت است که بخش عمده‌ای از تهدیدهای موجود علیه کشورها، برای تسلط بر زیرساخت‌های حیاتی سایبری آنان بوده است؛ بنابراین با توجه به آسیب‌پذیری‌های ذاتی موجود در فضای سایبری و روند روبه‌رشد مهاجرت از دنیای سنتی به این فضا، ریسک سامانه‌های مبتنی بر فناوری اطلاعات که برای کشورها حیاتی هستند، افزایش یافته است. براین‌اساس ارتقای پایداری عملیاتی و امنیت و مصون‌سازی زیرساخت‌ها به‌ویژه مراکز حیاتی و حساس برای کشورها بسیار حائز اهمیت تلقی می‌شود [۱].

در این پژوهش، یک روش تشخیص نفوذ چندکلاسه[1] پیشنهاد شده است. این روش از انتخاب ویژگی با الگوریتم سنجاقک[2] استفاده می‌کند. الگوریتم سنجاقک که برای اولین بار در سال ۲۰۱۵ ارائه شده است، در مسائل سخت فاقد[3] راه‌حل به کار گرفته می‌شود [۲]. همچنین، در این پژوهش به‌منظور تشخیص و دسته‌بندی حملات ترافیک شبکه، از الگوریتم جنگل تصادفی[4] استفاده شده است. انتخاب این روش، پس از اعمال روش‌های مختلف و آزمون آن‌ها، به دلیل عملکرد مناسب و میزان صحت، دقت و سرعت بالاتر، بوده است. در این تحقیق از مجموعه‌های داده‌های KDD-99 استفاده شده است سیستم موردتحقیق، علاوه بر تشخیص حمله یا غیرحمله بودن ترافیک وارده، توانایی تشخیص نوع حملات را نیز دارد.

در بخش دوم مقاله، پژوهش‌های پیشین بررسی خواهند شد و در بخش سوم الگوریتم‌های روش پیشنهادی معرفی می‌شوند. سپس نتایج تحقیق در بخش چهارم بیان خواهد شد و بخش پنجم به جمع‌بندی روش پیشنهادی می‌پردازد.

۲- پژوهش‌های پیشین

سیستم‌های تشخیص نفوذ را از نظر نوع فناوری تشخیص، می‌توان به روش‌های تشخیص آماری، روش‌های تشخیص مبتنی بر دانش و روش‌های مبتنی بر یادگیری ماشین تقسیم نمود.

* رایانامه نویسنده مسئول: Tanha@tabrizu.ac.ir

[1] Multi-Class
[2] Dragonfly Algorithm
[3] NP-Hard
[4] Random Forest

ناشر: دانشگاه جامع امام حسین (ع)

استفاده از تکنیک‌های یادگیری ماشین برای تشخیص ناهنجاری، مبتنی بر ساختن مدلی است که توسط آموزش روی مجموعه‌داده‌های قبلی به دست می‌آید. چنین مدلی قابلیت تعمیم روی نمونه‌های بعدی را دارد و می‌تواند با کارایی بالایی برای دسته‌بندی نمونه‌های جدید مورداستفاده قرار گیرد [۳].

در منبع شماره [۴] از یک روش ترکیبی برای تشخیص نفوذ استفاده‌شده است. در این پژوهش از ترکیب درخت تصمیم و جنگل تصادفی استفاده‌شده که توانایی انجام رگرسیون و طبقه‌بندی را دارد. سادگی برنامه‌نویسی، تمایل به غیرخطی بودن و محدود کردن شرایط با استفاده از درخت تصمیم ممکن است. مقدار صحت به‌دست‌آمده در این روش ۰/۹۷ می‌باشد.

در منبع [۵] ابتدا داده‌ها با استفاده از درخت سلسله‌مراتبی به دسته‌هایی تقسیم شده‌اند، سپس با استفاده از الگوریتم SVM داده‌های نرمال و مشکوک به حمله، تشخیص داده می‌شود. صحت در این مقاله مقدار ۰/۹۶۷۷ محاسبه شده است.

در سال ۲۰۲۰ پژوهشگر [۶]، برای تشخیص نفوذ، از روش یادگیری CNN و DNN روی مجموعه‌داده NSL-KDD به‌صورت چند کلاس استفاده کرده‌اند. حداکثر مقادیر به‌دست‌آمده در این بررسی، صحت ۰/۷۷۶۹، دقت ۰/۷۳۹۲ ، بازخوانی ۰/۵۲۸۶ و F1-Score برابر با ۰/۵۳۵۲ برای CNN بوده است.

در پژوهشی دیگر در سال ۲۰۱۹، از روش تشخیص ویژگی‌ها به‌منظور کشف نفوذ استفاده شده است. در این روش ابتدا با استفاده از روش فیلتر FGLCC و خوشه‌بندی، ویژگی‌ها تشخیص داده می‌شوند، سپس با استفاده از روش درخت تصمیم‌گیری در ارتباط با داده‌های عادی و یا نفوذ تصمیم‌گیری انجام می‌شود. در این روش از مجموعه‌داده KDD استفاده‌شده و دقت نهایی ۰/۹۵۰۳ درصد گزارش شده است [۷].

در مقاله‌ای دیگر، چوداری و همکارانشان [۸] از یک شبکه عصبی عمیق بر روی مجموعه‌داده KDD-99 و دو مجموعه‌داده دیگر به‌صورت باینری استفاده کرده‌اند. در این پژوهش مجموعه‌داده KDD-99 و دو مجموعه‌داده دیگر موردبررسی قرار گرفته و مقدار صحت نهایی در این روش ۰/۹۶۰۳ درصد محاسبه شده است. لازم به ذکر است که این پژوهش، توانایی تشخیص نفوذ و عدم نفوذ را در سیستم داشته و نوع حمله در آن تشخیص داده نمی‌شود.

در سال ۲۰۱۵ در یک پژوهش دیگر [۹]، از روش یادگیری STL[1] بر روی مجموعه‌داده NSL-KDD هم به‌صورت باینری و هم چند کلاس استفاده کرده است. یادگیری STL نوعی یادگیری ماشین است که توسط محققان استنفورد در سال ۲۰۰۷ معرفی شده است و ترکیبی از دو نوع یادگیری با برچسب و یادگیری با داده‌های بدون برچسب از کلاس‌های دیگر می‌باشد. نتایج پژوهش فوق با چند روش دیگر مقایسه گردیده و حداکثر مقدار صحت ۰/۹۸۸۴ درصد مربوط به STL گزارش شده است.

چنانچه مشاهده می‌شود، در برخی مقالات تشخیص نفوذ به‌صورت باینری بررسی شده‌اند و توانایی تشخیص نوع حمله را نداشته‌اند. همچنین در اکثر مقالات فقط معیار صحت[2] محاسبه شده است.

به‌منظور رفع موارد فوق، در پژوهش حاضر روشی موردبررسی قرار گرفته که بتواند نوع حملات را نیز مشخص نماید. همچنین معیارهای دقت، بازخوانی و F1-Score نیز محاسبه شده و با چندین الگوریتم یادگیری ماشین و عمیق مورد مقایسه قرار گیرد. در این پژوهش از یک روش جدید انتخاب ویژگی استفاده شده تا حجم محاسبات و زمان اجرای روش تا اندازه زیادی کاهش یابد. در ضمن، چنانچه در بخش چهار مشاهده خواهد شد نتایج روش پیشنهادی از نتایج حاصل از اکثر پژوهش‌های انجام‌شده بهتر بوده است.

۳- روش پیشنهادی

سیستم‌های تشخیص نفوذ که با رویکردهای معمول یادگیری ماشین ارائه می‌شود، اغلب دارای مشکلاتی هستند، مانند مشکل انتخاب ویژگی توسط افراد خبره یا مشکل کاهش ابعاد برای حذف نویز در برخورد با داده‌های با ابعاد بسیار زیاد که معمولاً منجر به حذف اطلاعات قابل‌توجهی از مجموعه‌داده‌ها می‌شود. همچنین در مواردی که متغیرهای ترافیک شبکه پیچیده و تعداد آن‌ها بسیار زیاد باشد.

برای کاهش مشکلات فوق، در این پژوهش روشی پیشنهاد شده که ویژگی‌ها با استفاده از یک الگوریتم فراابتکاری[3] به نام الگوریتم سنجاقک انتخاب می‌شوند. همچنین دسته‌بندی حملات با استفاده از الگوریتم جنگل تصادفی صورت می‌پذیرد. در ادامه این بخش، ابتدا مجموعه‌داده‌های مورداستفاده معرفی و سپس الگوریتم سنجاقک و جنگل تصادفی تشریح و در نهایت الگوریتم مدل پیشنهادی معرفی شده است.

۳-۱- معرفی مجموعه‌داده

در این پژوهش، از مجموعه‌داده KDD-99 استفاده شده است. این مجموعه‌داده یکی از پراستفاده‌ترین مجموعه‌داده‌ها در زمینه تشخیص نفوذ است که به‌صورت عمومی در دسترس می‌باشد. KDD-99 که توسط آزمایشگاه ام‌آی‌تی لینکلن جمع‌آوری شده،

[1] Self-taught learning

[2] Accuracy

[3] Meta-heuristic Algorithms

یک نسخه از مجموعه‌داده DARPA-1998 می‌باشد. این مجموعه‌داده شامل حدود ۵ میلیون رکورد می‌باشد. هر نمونه داده، دارای ۴۱ ویژگی می‌باشد که تعداد ۳۸ ویژگی به‌صورت عددی بوده و ۳ ویژگی به‌صورت غیرعددی می‌باشند [۱۰].

این مجموعه‌داده برای تشخیص حملات ممانعت از سرویس[1]، جستجوگر[2]، U2R[3] و R2L[4] مورداستفاده قرار می‌گیرد.

۲-۳- الگوریتم انتخاب ویژگی سنجاقک

مشکل بعضی از کاربردهای یادگیری ماشین، تعداد زیاد ویژگی‌هایی می‌باشد که علی‌رغم نداشتن ارزش اطلاعاتی مناسب، بار محاسباتی بالایی را به سیستم تحمیل می‌کنند. اگرچه امروزه با ظهور کامپیوترهای سریع و منابع ذخیره‌سازی بزرگ این مشکل به چشم نمی‌آید ولی از طرف دیگر، مجموعه‌های داده‌ای بسیار بزرگ برای مسائل جدید باعث شده است که همچنان پیدا کردن یک الگوریتم دقیق‌تر برای انتخاب ویژگی، اهمیت داشته باشد.

از انواع الگوریتم‌های انتخاب ویژگی که در پژوهش‌ها از آن‌ها استفاده می‌شود، می‌توان به انتخاب ویژگی با استفاده از الگوریتم کلونی زنبورعسل[5]، الگوریتم ژنتیک[6]، الگوریتم کرم شب‌تاب[7] و الگوریتم فاخته[8] اشاره نمود [۱۱].

انتخاب ویژگی در این پژوهش با استفاده از الگوریتم سنجاقک می‌باشد. این روش پس از مطالعه بر روی مقالات مربوط به الگوریتم سنجاقک و مقایسه با سایر الگوریتم‌های انتخاب ویژگی، برگزیده شده است.

الگوریتم فراابتکاری سنجاقک در سال ۲۰۱۵ به‌عنوان یک الگوریتم بهینه‌سازی پیشنهاد شده است. این الگوریتم که بر اساس رفتار سنجاقک طبیعی طراحی شده است. رفتار هوشمندانه یک سنجاقک بر پنج اصل استوار می‌باشد. این اصول عبارت‌اند از: «اجتناب از برخورد با سایر افراد همسایه»، «تنظیم سرعت با توجه به موقعیت سایر افراد همسایه»، «تمایل سنجاقک به سمت مرکز ثقل همسایه‌ها»، «جذب به سمت منبع غذایی» و «فرار از دشمن». براین‌اساس در الگوریتم سنجاقک پنج تابع تعریف شده و مورداستفاده قرار گرفته است [۲]:

۱- تابع جداسازی[9]: زمانی اتفاق می‌افتد که سنجاقک‌ها برای جلوگیری از برخورد با همسایگان از آن تبعیت می‌کنند. رابطه ریاضی این تابع به‌صورت (۱) می‌باشد.

$$S_i = -\sum_{j=1}^{N} X - X_j \quad (۱)$$

که X موقعیت فعلی سنجاقک ، Xj موقعیت همسایه X بوده و N اندازه فضای موجود است.

۲- تابع سرعت[10]: این تابع سرعت سنجاقک‌ها را با توجه به سنجاقک‌های همسایه به‌صورت (۲) محاسبه می‌کند.

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \quad (۲)$$

که Vj سرعت j امین سنجاقک می‌باشد.

۳- تابع انسجام[11]: این تابع به‌صورت (۳) پیوستگی همسایه‌ها را محاسبه می‌کند.

$$C_i = \frac{\sum_{j=1}^{N} X_i}{N} - X \quad (۳)$$

که در آن Xj موقعیت سنجاقک همسایه X را نشان می‌دهد.

۴- تابع تمایل[12]: این تابع، جذب سنجاقک به سمت منبع غذایی را نشان می‌دهد که به‌صورت (۴) محاسبه می‌شود.

$$F_i = X+ - X \quad (۴)$$

که X+ موقعیت منبع غذایی را نشان می‌دهد.

۵- تابع فرار[13]: رفتار طبیعی که هر سنجاقک برای زنده ماندن در مقابل نفوذ دشمن انجام می‌دهد. این تابع را در (۵) می‌توان مشاهده کرد.

$$E_i = X- + X \quad (۵)$$

که در آن X- موقعیت دشمن را بیان می‌کند.

در الگوریتم سنجاقک، ابتدا بردارهای موقعیت و گام، به‌طور تصادفی و با توجه به حد پایین و بالای متغیرها مقداردهی می‌شوند. سپس در هر تکرار، بهترین موقعیت و گام سنجاقک‌ها، پی‌درپی به‌روزرسانی می‌شوند. برای به‌روزرسانی بردار موقعیت سنجاقک، از بردار گام (مرحله) (ΔX) و بردار موقعیت فعلی، استفاده می‌شود. بردار گام، جهت حرکت سنجاقک را نشان می‌دهد که به‌صورت (۶) محاسبه می‌شود.

$$\Delta Xt+1 = (sSi + aAi + cCi + fFi + eEi) + wXt \quad (۶)$$

که پارامترهای e,w,a,c,s,f بردارهای وزن می‌باشند. همچنین

[1] Denial of service(DOS)
[2] Probe
[3] User to Root
[4] Remote to Local
[5] Bee optimization Algorithm
[6] Genetic Algorithm
[7] Firefly optimization Algorithm
[8] Cuckoo optimization Algorithm
[9] Separation
[10] Alignment
[11] Cohesion
[12] Attraction
[13] Escape

موقعیت سنجاقک‌ها با استفاده از (۷) به‌روزرسانی می‌شود.

$$X_{t+1} - X_t = \Delta X_{t+1} \quad (۷)$$

که در آن پارامتر t تعداد تکرارها را نشان می‌دهد.

در فرایند انتخاب ویژگی، یک سنجاقک ویژگی‌هایی را به‌صورت تصادفی انتخاب و با استفاده از الگوریتم KNN، برای هر نمونه با ویژگی‌های جدید، دسته موردنظر را پیش‌بینی می‌کند. شبیه‌ترین نمونه‌ها در این پیش‌بینی، با استفاده از محاسبه فاصله منهتن (بین ترافیک برچسب نزده و برچسب زده شده) انجام می‌گردد. سپس با توجه به وضعیت همسایه‌ها، در خصوص دسته موردنظر تصمیم گیری می‌کند.

این پیش‌بینی، با ویژگی‌های انتخاب شده برای تمام نمونه‌ها انجام می‌شود سپس مقدار نمونه‌های پیش‌بینی شده، با مقدار برچسب واقعی مقایسه می‌شود اگر برابر باشند، خروجی (۱) و در غیر این صورت خروجی (۰) را برمی‌گرداند. این عملیات برای تمام نمونه‌ها صورت می‌پذیرد و تعداد آن‌ها محاسبه و سنجاقکی که مجموعه نمونه‌های درست بیشتری را پیش‌بینی کرده باشد به‌عنوان BF [1] انتخاب می‌شود.

هنگامی‌که هیچ راه‌حل همسایه‌ای وجود نداشته باشد، برای افزایش تصادفی بودن، سنجاقک‌ها باید از یک پیاده‌روی تصادفی برای پرواز در سراسر فضای جستجو استفاده کنند.

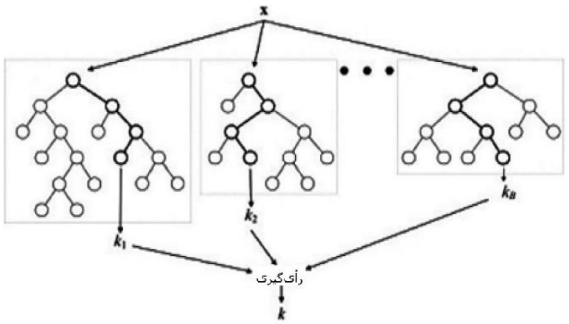
در مرحله بعد مطابق روابط (۶) و (۷) سنجاقک‌ها جابه‌جا و مجدداً BF محاسبه شده و با مقادیر قبلی مقایسه می‌گردد. الگوریتم به تعداد مشخص (که توسط محقق انتخاب می‌گردد) تکرار شده و سنجاقکی که دارای بالاترین BF است انتخاب می‌شود و ویژگی‌هایی که آن سنجاقک انتخاب نموده ویژگی‌های بهینه خواهند بود [۱۲].

۳-۳- الگوریتم جنگل تصادفی

الگوریتم جنگل تصادفی متشکل از چندین الگوریتم درخت تصمیم می‌باشد. به‌عبارت‌دیگر در این الگوریتم، برای آموزش داده‌ها چندین درخت ساخته می‌شود که در هرکدام از آن‌ها ترتیب قرار گرفتن ویژگی‌ها در اعماق درخت، به‌صورت تصادفی تعیین می‌گردد.

جنگل تصادفی پیش‌بینی را از هر درخت و بر اساس اکثریت آرا پیش‌بینی می‌کند و نتیجه نهایی را به‌عنوان خروجی در نظر می‌گیرد. تعداد بیشتر درختان در جنگل منجر به‌دقت بالاتری می‌شود.

یکی از پارامترهای مهم و تأثیرگذار در جنگل تصادفی، تعداد درخت‌های تولید شده می‌باشد. از مزایای الگوریتم جنگل تصادفی می‌توان به پاسخگویی بهتر در داده‌های با حجم بالا، پایداری الگوریتم، قابل‌استفاده برای رگرسیون و دسته‌بندی، محدود بودن تعداد پارامترها و آسان بودن استفاده از آن نام برد. عملکرد یک الگوریتم جنگل تصادفی در شکل (۱) مشاهده می‌شود [۱۳].



شکل (۱). نحوه عملکرد جنگل تصادفی

لازم به ذکر است در پژوهش حاضر به‌منظور انتخاب الگوریتم دسته‌بندی، اکثر الگوریتم‌های یادگیری ماشین و یادگیری عمیق مورد آزمایش قرار گرفته و پس از مقایسه نتایج، الگوریتم جنگل تصادفی به‌عنوان سریع‌ترین و بهترین الگوریتم انتخاب شده است. شبه کد مدل پیشنهاد شده در این پژوهش مطابق نمودار شماره (۱) می‌باشد.

Initialize:
The network/Server/Firewall traffic dataset
preprocessing:
Remove the zero features
Delete equal rows
Correct or delete rows with lost and infinity data
Grouping and quantifying attacks
Quantifying data
Normalization operation (Using Min-Max)
Feature selection:
Initialize the dragonflies population Xi (i = 1, 2,., n)
Initialize step vectors ΔXi (i = 1, 2, ..., n)
Select Random variables
while the end condition is not satisfied
for each observation
) Using knn(Forecast the label
Calculate The $fittnes$ function مطابق رابطه(۸)
End for
Update w, s, a, c, f, and e
Calculate S, A, C, F, and E مطابق روابط (۱) تا (۵)
Update neighbouring radius
neighbouring dragonfly If a dragonfly has at least one
Update velocity vector مطابق رابطه (۶)

[1] Best Fittness

```
Update position vector مطابق رابطه (۷)
Else
Update the position vector using L´evy flight
End if
Check and correct the new positions based on the boundaries of variables
Calculate the values of all dragonflies
Name the biggest fittnes function , best Fittness
.End while
Choose the features of the best dragonfly
Classify types of attacks using Random forest Algorithm
End.
```

نمودار (۱). شبه کد روش پیشنهادی

چنانچه در نمودار (۱) مشاهده می‌شود، داده‌های خام پس از پیش‌پردازش و یکسان کردن مقیاس داده‌ها از طریق روش کمترین - بیشترین[1]، با استفاده از الگوریتم سنجاقک انتخاب ویژگی شده و سپس نتایج از طریق الگوریتم جنگل تصادفی دسته‌بندی می‌شوند.

۴- آزمایش‌ها و نتایج

این بخش شامل نتایج پیش‌پردازش داده‌ها، ویژگی‌های انتخاب شده و نتایج عملیات دسته‌بندی حملات می‌باشد و در انتها مقایسه نتایج روش پیشنهادی با سایر روش‌های موجود در این زمینه بیان شده است. لازم به ذکر است سیستم رایانه مورداستفاده در این پژوهش، دارای پردازنده Core i5 و مقدار ۸ گیگابایت RAM بوده و از ابزارهای پایتون در محیط Colab و Matlab 2020 b استفاده شده است.

در این پژوهش از معیارهای صحت[2]، دقت[3]، میزان بازخوانی[4]، F1-Score و همچنین معیار زمان برای ارزیابی الگوریتم‌ها استفاده شده است.

معیار صحت[5]: این معیار تعداد پیش‌بینی درست از تعداد کل پیش‌بینی‌ها را نشان می‌دهد و با رابطه (۸) محاسبه می‌گردد.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \qquad (۸)$$

معیار دقت[6]: معیاری است که نشان می‌دهد در نمونه‌های مختلف، پیش‌بینی‌ها به چه میزانی واقعاً درست بوده‌اند. این معیار به‌صورت (۹) قابل‌محاسبه می‌باشد.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (۹)$$

معیار فراخوانی[7]: این معیار، درصد نفوذهای واقعی پوشش داده شده توسط سیستم را مشخص می‌کند. این معیار با رابطه (۱۰) محاسبه می‌شود.

$$\text{Recall} = \frac{Tp}{TP+FN} \qquad (۱۰)$$

معیار F1-score نیز میانگین دقت و فراخوانی می‌باشد که با رابطه (۱۱) محاسبه می‌گردد.

$$F1\_Score = \frac{2.\text{Recall}.\text{Precision}}{\text{Recall}+\text{Precision}} \qquad (۱۱)$$

TP[8]- تعداد حملاتی که به‌عنوان حمله طبقه‌بندی شدند.

TN[9]- تعداد رکوردهای نرمال که به‌عنوان نرمال طبقه‌بندی شدند.

FP[10] - تعداد حملاتی که به‌عنوان نرمال طبقه‌بندی شده‌اند.

FN[11] - تعداد رکوردهای نرمال که به‌عنوان حمله طبقه‌بندی شدند [۱۴].

۴-۱- پیش‌پردازش داده‌ها

در مرحله پیش‌پردازش داده‌ها، ابتدا ویژگی‌هایی که تمام مقادیر آن‌ها صفر بود و همچنین داده‌های گم‌شده[12] و داده‌هایی که مقداری برای آن‌ها ثبت نشده بود (دارای سلول‌های خالی) حذف شدند، سپس ویژگی‌های protocol_type و service و flag که مقدار آن‌ها غیرعددی می‌باشد، به مقادیر عددی تبدیل شدند. در ادامه ترافیک به‌صورت چهار گروه حمله و یک گروه نرمال عددی شدند. سپس برای یکسان کردن مقیاس داده‌ها و نرمال‌سازی[13] آن‌ها، از روش کمترین- بیشترین استفاده شد.

در این پژوهش پس از انجام مراحل پیش‌پردازش، به‌دلیل کم بودن مشاهدات در برخی از کلاس‌ها، از وزن‌دهی به کلاس‌ها استفاده شده است. در وزن‌دهی کلاس‌ها، به نسبت تعداد مشاهدات هر کلاس، به آن‌ها وزن اختصاص می‌یابد تا کلاس‌های اقلیت نیز در پیش‌بینی نقش داشته باشند. لازم به ذکر است در برخی از پژوهش‌ها روش‌هایی مانند نمونه‌گیری افزایشی یا کاهشی به کار برده می‌شوند، لیکن در این پژوهش از این روش‌ها استفاده نشده است.

[1] Min-Max
[2] Accuracy
[3] Precision
[4] Recall
[5] Accuracy
[6] Precision
[7] Recall
[8] True Positive
[9] True Negative
[10] False Positive
[11] False Negative
[12] Missing Data
[13] Normalization

۲-۴- نتایج الگوریتم‌های یادگیری بر روی داده‌های اولیه (قبل از اعمال روش پیشنهادی)

در این مرحله، ابتدا داده‌های به‌دست‌آمده، به دو گروه ویژگی‌ها (X) و برچسب‌ها (Y) تقسیم شدند و سپس تعداد یک درصد از سطرهای مجموعه‌داده به‌طور تصادفی برگزیده شدند. این انتخاب به شکلی بوده که به‌تناسب، ترکیبی از تمام انواع برچسب‌ها انتخاب شده باشند. این برچسب‌ها به‌صورت X_train، y_train، X_test و y_test نام‌گذاری گردیدند.

در مرحله بعد الگوریتم‌های یادگیری بر روی X_train و y_train مجموعه‌داده اعمال و نتایج صحت، دقت، میانگین تقریبی، بازخوانی و زمان اجرا محاسبه گردید که در جداول شماره (۱) تا (۴) درج شده‌اند.

جدول (۱). نتایج صحت الگوریتم‌ها قبل از انتخاب ویژگی

الگوریتم	LR	DT	KNN	RF	GB
صحت	۰/۸۴۷۹	۰/۷۰۲۴	۰/۸۹۱۳	۰/۹۰۰۲	۰/۸۳۱۹

چنانچه در جدول شماره (۱) دیده می‌شود مقدار صحت مربوط به الگوریتم جنگل تصادفی بیشتر از بقیه الگوریتم‌های یادگیری ماشین مورد آزمایش محاسبه شده است. در روش‌های عمیق بازگشتی RNN، GRU و LSTM نیز آرگومان‌های مختلف آزمایش و بهترین مقادیر به‌صورت جدول شماره (۲) به‌دست‌آمده‌اند.

جدول (۲). نتایج صحت روش‌های عمیق بازگشتی قبل از مراحل انتخاب ویژگی

الگوریتم	RNN	GRU	LSTM
لایه پنهان	۲	۲	۲
تعداد نود	۵۱۲-۵۱۲	۵۱۲-۵۱۲	۵۱۲-۵۱۲
نرخ یادگیری	3e-4	3e-4	3e-4
تعداد تکرار	۵	۵	۵
فعال‌سازی	Relu – SoftMax	Relu – SoftMax	Relu – SoftMax
Batch	۶۴	۶۴	۶۴
Verbose	۲	۲	۲
ACC	۰/۷۵	۰/۷۱۹	۰/۸

چنانچه از جدول شماره (۲) و مقایسه آن با جدول شماره (۱) مشاهده می‌شود، صحت مربوط به جنگل تصادفی بهتر می‌باشد. مدت‌زمان اجرای الگوریتم‌های موردبررسی قبل از اعمال روش پیشنهادی در جدول شماره (۳) نشان داده شده است.

جدول (۳). زمان اجرای الگوریتم‌ها قبل از انتخاب ویژگی

الگوریتم	زمان آموزش	زمان آزمایش
LR	۳۲	۰/۱۲
DT	۴/۵	۰/۰۹
KNN	۱۰/۱۲	۰/۸۱
RF	۳/۲	۰/۰۱
GB	۷/۸	۰/۱۲
RNN	۱۳۹	۱
GRU	۲۸۵	۱
LSTM	۴۱۸	۱

در جدول شماره (۴) میانگین حسابی نتایج معیارهای F1-Score، بازخوانی و دقت روش‌های بررسی شده، قبل از انتخاب ویژگی مشاهده می‌شود.

جدول (۴). میانگین نتایج F1-Score، بازخوانی و دقت الگوریتم‌ها قبل از انتخاب ویژگی

الگوریتم	میانگین F1-Score	میانگین بازخوانی	میانگین دقت
LR	۰/۳۵۴	۰/۳۹	۰/۴۷
DT	۰/۵۱	۰/۷۶	۰/۳۹
KNN	۰/۸	۰/۷۱	۰/۸۷
RF	۰/۷۶	۰/۸۲	۰/۸۳۷
GB	۰/۷۵	۰/۷۸	۰/۸۰۹
RNN	۰/۴	۰/۷۹	۰/۵۶
GRU	۰/۷۶	۰/۶۹	۰/۵۱
LSTM	۰/۶۵	۰/۸	۰/۴۵

روش پیشنهادی در این پژوهش استفاده از انتخاب ویژگی سنجاقک در مجموعه‌داده می‌باشد که به‌صورت زیر انجام شده است.

۳-۴- نتایج انتخاب ویژگی سنجاقک

اجرای عملیات انتخاب ویژگی سنجاقک، در مجموعه‌داده، پس از ۴ بار تکرار و تعداد ۵ سنجاقک و انتخاب KNN=۵، موجب انتخاب ۱۹ ویژگی با بیشترین اولویت شد که در مقایسه با سایر ویژگی‌ها با دقت ۹۹/۰۸ برنده شدند. زمان موردنیاز برای اجرای این الگوریتم، حدود ۶۹۹ ثانیه بوده است. گزارش انتخاب ویژگی‌ها در جدول شماره (۵) نشان داده شده‌اند.

جدول (۵). ویژگی‌های انتخاب شده توسط الگوریتم سنجاقک

ویژگی های انتخاب شده	F1	F2	F3	F4	F5
	۱	۲	۳	۴	۸
F6	F7	F8	F9	F10	F11
۹	۱۰	۱۳	۱۴	۱۵	۱۷
F12	F13	F14	F15	F16	F17
۱۸	۲۰	۲۱	۲۳	۲۷	۳۳
F18	F19				
۳۴	۳۸				

در جدول (۵) ویژگی nام مجموعه‌داده جدید، با متغیر Fn نشان داده شده و عدد متناظر با آن شماره ویژگی در مجموعه‌داده اولیه می‌باشد.

۴-۴- نتایج الگوریتم‌های یادگیری (پس از انتخاب ویژگی)

در این مرحله، الگوریتم انتخاب ویژگی سنجاقک بر روی داده‌ها اعمال شده است. چنانچه در جدول شماره (۵) دیده می‌شود، تعداد ویژگی‌ها به ۱۹ ویژگی کاهش‌یافته است. الگوریتم‌های یادگیری بر روی داده‌های نهایی اعمال و نتیجه روش‌ها در جداول شماره (۶) تا (۹) ثبت گردیده است.

جدول (۶). نتایج صحت پس از مراحل انتخاب ویژگی

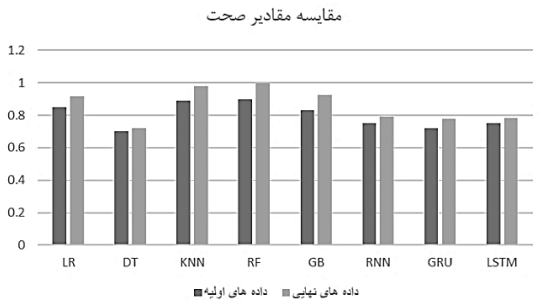
الگوریتم	LR	DT	KNN	روش پیشنهادی	GB
صحت	۰/۹۱۵۶	۰/۷۲۰۱	۰/۹۷۹	۰/۹۹۶۷	۰/۹۲۲۹

همچنین، الگوریتم‌های یادگیری عمیق نیز با داده‌های جدید آزمایش شدند، در این مرحله برای آزمایش سیستم، از تعداد ۱۰ تکرار استفاده شد تا اگر پتانسیل بهبود وجود دارد مشخص گردد. لیکن مشاهده شد که با تکرارهای زیاد نیز، مقدار صحت تااندازه‌ای افزایش یافته و پس از آن مجدداً افت پیدا می‌کند. این نتایج در جدول شماره (۷) نشان داده شده است.

جدول (۷). نتایج صحت الگوریتم‌های عمیق پس از مراحل انتخاب ویژگی

الگوریتم	RNN	GRU	LSTM
صحت	۰/۷۹	۰/۷۷۶	۰/۷۸

مقایسه نتایج صحت الگوریتم‌ها با داده‌های اولیه و داده‌های نهایی (روش پیشنهادی) در نمودار شماره (۲) نشان داده شده است.



نمودار (۲). مقایسه نتایج صحت الگوریتم‌ها با داده‌های اولیه و داده‌های نهایی

چنانچه در نمودار (۲) مشاهده می‌شود دسته‌بندی جنگل تصادفی با استفاده از مجموعه‌داده انتخاب شده، بالاترین صحت را به دست آورده است. در جدول شماره (۸)، زمان اجرای الگوریتم‌ها پس از انتخاب ویژگی مشاهده می‌شود.

جدول (۸). زمان اجرای الگوریتم‌ها پس از انتخاب ویژگی

الگوریتم	زمان آموزش	زمان آزمایش
LR	۲۷	۰/۰۸
DT	۳/۸	۰/۰۳
KNN	۷/۵	۰/۵۹
روش پیشنهادی	**۱/۲۸**	**۰/۰۰۹**
GB	۶/۹	۰/۰۲
RNN	۶۹	۱
GRU	۳۱۸	۱
LSTM	۴۰۱	۱

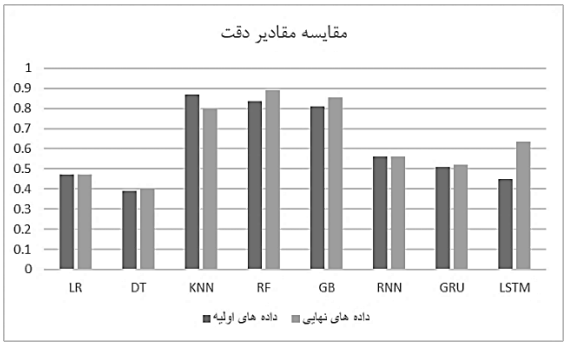
در جدول شماره (۹) میانگین مقادیر F1-Score، بازخوانی و دقت الگوریتم‌های یادگیری پس از انتخاب ویژگی به ثبت رسیده است.

جدول (۹). نتایج میانگین بازخوانی، F1-Score و دقت الگوریتم‌های یادگیری پس از انتخاب ویژگی

الگوریتم	میانگین F1-Score	میانگین بازخوانی	میانگین دقت
LR	۰/۴۱۲	۰/۳۹	۰/۴۷
DT	۰/۵۴۸	۰/۸۸	۰/۴
KNN	۰/۸۴	۰/۸۵۵	۰/۸
روش پیشنهادی	**۰/۸۷۱**	**۰/۸۵۶**	**۰/۸۹**
GB	۰/۷۷	۰/۸	۰/۸۵۴
RNN	۰/۵۷۷	۰/۸۱۳	۰/۵۶
GRU	۰/۷۸	۰/۸۱۹	۰/۵۲۱
LSTM	۰/۶۶	۰/۸۷۳	۰/۶۳۵

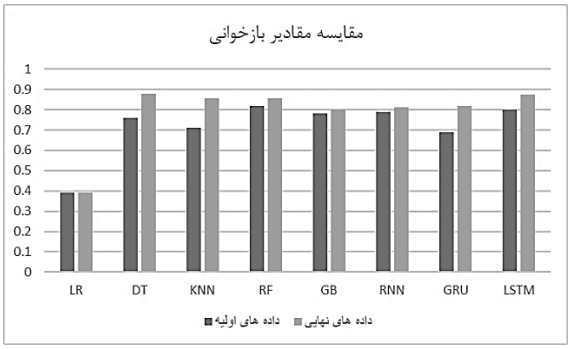
چنانچه در جدول شماره (۹) مشاهده می‌شود مقادیر F1-Score، بازخوانی و دقت در اکثر الگوریتم‌ها علی‌الخصوص در الگوریتم جنگل تصادفی (روش پیشنهادی) افزایش داشته است.

در نمودار شماره (۳) نتایج به‌دست‌آمده از مقادیر دقت الگوریتم‌های اجرا شده با داده‌های اولیه (پس از مراحل پیش‌پردازش) و داده‌های نهایی (پس از انتخاب ویژگی) مقایسه شده‌اند.



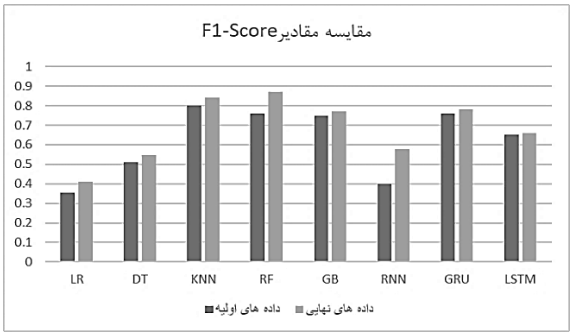
نمودار (۳). مقایسه نتایج دقت الگوریتم‌ها با داده‌های اولیه و داده‌های نهایی

در نمودار شماره (۳) مشاهده می‌شود که مقدار دقت در روش پیشنهادی نسبت به داده‌های قبل از انتخاب ویژگی بهبودیافته و نسبت به سایر روش‌ها نیز مقدار بالاتری را به خود اختصاص داده است. نمودار شماره (۴) نتایج مقادیر بازخوانی الگوریتم‌ها را قبل و بعد از مراحل انتخاب ویژگی نشان می‌دهد.



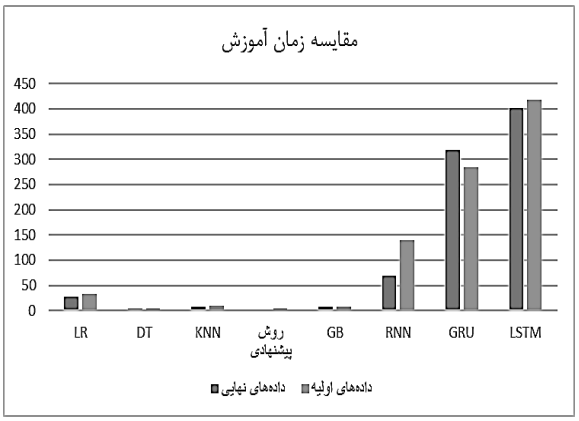
نمودار (۴). مقایسه نتایج بازخوانی الگوریتم‌ها با داده‌های اولیه و داده‌های نهایی

در نمودار شماره (۴) دیده می‌شود که مقدار بازخوانی روش پیشنهادی نسبت به داده‌های اولیه رشد داشته است. البته این مقدار، نسبت به سایر الگوریتم‌ها بالاترین مقدار را به خود اختصاص نداده است. نمودار شماره (۵) مقادیر F1-Score الگوریتم‌ها قبل و بعد از مراحل انتخاب ویژگی را نشان می‌دهد.



نمودار (۵). مقایسه نتایج F1-Score با داده‌های اولیه و داده‌های نهایی

چنانچه در نمودار شماره (۵) دیده می‌شود، مقدار F1-Score در روش پیشنهادی نسبت به سایر روش‌ها بالاتر می‌باشد و این روش نسبت به داده‌های اولیه، افزایش داشته است. زمان اجرای الگوریتم‌ها در نمودار شماره (۶) مورد مقایسه قرار گرفته است.



نمودار (۶). مقایسه مدت‌زمان اجرا با داده‌های اولیه و داده‌های نهایی

چنانچه در نمودار (۶) معلوم است، زمان آموزش روش پیشنهادی نسبت به بقیه الگوریتم‌ها کمتر بوده و نسبت به داده‌های اولیه کاهش‌یافته است.

۵-۴- مقایسه نتایج روش پیشنهادی با پژوهش‌های پیشین

به‌منظور مقایسه نتایج روش پیشنهادی با نتایج پژوهش‌های مشابه، برخی از این پژوهش‌ها انتخاب و نتایج کسب شده در جدول (۱۰) موردمطالعه قرار گرفته‌اند.

منبع شماره [۴] که از ترکیب درخت تصمیم و جنگل تصادفی برای تشخیص نفوذ استفاده نموده است، معیار صحت را ۰/۹۷ گزارش نموده است که در مقایسه با روش پیشنهاد شده در پژوهش حاضر مقدار کمتری می‌باشد، در منبع فوق معیارهای دیگر موردبررسی قرار نگرفته است.

مقدار صحت در مقاله [۵] ۰/۹۶۷۷ محاسبه شده است. در این پژوهش که از الگوریتم SVM استفاده شده، معیارهای دقت، بازخوانی، میانگین تقریبی و سایر مقادیر محاسبه نشده‌اند.

در پژوهش [۶] که یادگیری CNN به‌صورت چند کلاس به کار گرفته شده است، مقادیر صحت ۰/۷۷۶۹ ، دقت ۰/۷۳۹۲، بازخوانی ۰/۵۲۸۶ و F1-Score ۰/۵۳۵۲ را به‌عنوان حداکثر معیارها گزارش شده است. با مقایسه نتایج پژوهش حاضر و این مقادیر، مشاهده می‌شود که تمامی مقادیر نسبت به مقاله فوق ارتقا یافته‌اند.

جدول (۱۰). مقایسه نتایج پژوهش با نتایج پژوهش‌های پیشین بر روی مجموعه‌داده‌های مشابه

شماره ارجاع	روش یادگیری	سال ارائه	مجموعه‌داده	معیار ارزیابی	مقدار
[۴]	DT-RF		KDD-CUP99	صحت	۰/۹۷
[۵]	SVM		KDD-CUP99	صحت	۰/۹۶۷۷
[۶]	CNN		NSL-KDD	صحت	۰/۷۷۶۹
				دقت	۰/۷۳۹۲
				بازخوانی	۰/۵۲۸۶
				F1-Score	۰/۵۳۵۲
[۷]	DT FGLCC		NSL-KDD	صحت	۰/۹۵۰۳
[۸]	DL		KDD-CUP99	صحت	۰/۹۶۳
[۹]	STL		NSL-KDD	صحت	۰/۹۸۸۴
روش پیشنهادی	**جنگل تصادفی و انتخاب ویژگی سنجاقک**		**KDD99**	**صحت**	**۰/۹۹۶۷**
				دقت	**۰/۸۹**
				بازخوانی	**۰/۸۵۶**
				F1-Score	**۰/۸۷۱**

در پژوهش [۷] که از روش فیلتر FGLCC و خوشه‌بندی، ویژگی‌ها تشخیص داده شده و با استفاده از الگوریتم درخت تصمیم دسته‌بندی انجام می‌شود، صحت نهایی ۰/۹۵۰۳ درصد گزارش شده است.

روش مورداستفاده در منبع [۸] شبکه عصبی عمیق[1] بوده که به‌صورت باینری بر روی مجموعه‌داده KDD99 و دو مجموعه‌داده دیگر استفاده شده است که مقدار صحت آن بر روی مجموعه‌داده فوق، ۰/۹۶۳ بوده که پایین‌تر از مقدار صحت در روش پیشنهاد شده در این پژوهش می‌باشد.

پژوهشگر در منبع [۹] از یادگیری STL [2] هم به‌صورت باینری و هم با پنج کلاس بر روی مجموعه‌داده NSL-KDD استفاده نموده است. حداکثر مقدار محاسبه شده برای معیار صحت ۰/۹۸۸۴ بوده که کمتر از مقدار صحت روش پیشنهادی بوده است.

چنانچه در جدول شماره (۱۰) مشاهده می‌شود اکثر پژوهش‌های مشابه، از معیار صحت به‌منظور ارزیابی استفاده نموده‌اند و معیارهای دیگر موردمحاسبه قرار نگرفته است. مقدار صحت در روش پیشنهادی، نسبت به پژوهش‌های عنوان‌شده، مقادیر بالاتری را کسب کرده است. همچنین میانگین صحت پژوهش‌های ذکرشده با مجموعه‌داده KDD99 و مشابه آن، مقدار ۰/۹۳۶۱ بوده است که از مقدار صحت روش پیشنهادی پایین‌تر می‌باشد.

۵- نتیجه‌گیری و پیشنهادها

در این پژوهش مدل روش پیشنهادی به‌عنوان یک مدل تشخیص نفوذ با استفاده از الگوریتم سنجاقک برای انتخاب ویژگی و الگوریتم جنگل تصادفی به‌منظور دسته‌بندی پیشنهاد شد. این روش بر روی مجموعه‌داده معتبر KDD-99 اعمال شده و با چندین روش شاخص یادگیری ماشین و یادگیری عمیق مقایسه گردیده است. نتایج معیارهای ارزیابی‌ها نشان می‌دهند که سیستم پیشنهادی، معیارهای صحت، دقت، بازخوانی و F1-Score بالاتری نسبت به سایر الگوریتم‌های اجرا شده، داشته و زمان کمتری نیز در اجرای سیستم صرف نموده است. به‌طوری که با دسته‌بندی الگوریتم جنگل تصادفی، پیش از اعمال انتخاب ویژگی، معیار صحت ۰/۹۰۰۲بوده و معیارهای دقت، F1-Score و بازخوانی، به‌ترتیب دارای میانگین ۰/۷۶و ۰/۸۲ و ۰/۸۳۷ بوده‌اند که با اعمال انتخاب ویژگی در روش پیشنهادی، معیار صحت به مقدار ۰/۹۹۶۷ ارتقا یافته و سایر مقادیر نیز به ترتیب به نتایج ۰/۸۹و ۰/۸۵۶و ۰/۸۷۱ رسیده‌اند. چنانچه ملاحظه می‌شود تمامی معیارها نسبت به مقادیر قبل از اعمال روش، ارتقا یافته‌اند. در ضمن با توجه به جدول (۱۰) نتایج روش پیشنهاد شده، نسبت به اکثر مطالعات پیشین افزایش داشته است.

همچنین، سرعت اجرای روش پیشنهادی نیز نسبت به قبل از تغییرات داده شده بر روی داده‌ها ارتقا یافته است. البته با توجه به متفاوت بودن سخت‌افزار مورداستفاده در سایر پژوهش‌ها، مقایسه زمان کدهای اجرا شده با آن‌ها قابل استناد نمی‌باشد، لیکن مقایسه زمان روش‌های مختلف بکار رفته در این پژوهش که در نمودار (۶) قابل‌مشاهده است، حاکی از سرعت بالای روش پیشنهادی نسبت به الگوریتم‌های دیگر می‌باشد.

[1] Deep neural Network
[2] Self-taught learning

در نهایت برای پژوهش‌های آینده، با توجه به عدم توازن در مجموعه‌داده‌های موجود، پیشنهاد می‌شود عملیات متوازن‌سازی داده‌ها بر روی مجموعه‌داده KDD-99 انجام شود. در ضمن با توجه به اهمیت موضوعِ تشخیص و پیش‌بینی هدف و مقصد مهاجم، توصیه می‌شود در تحقیقات آینده، سیستم تشخیص و پیش‌بینی مقاصد حمله (بر اساس نوع حملهٔ تشخیص داده شده در این پژوهش) موردمطالعه قرار گیرد. همچنین پیشنهاد می‌شود سیستم توصیه‌گرِ مدیر شبکه، بر اساس خروجی روش پیشنهادی و نقشه شبکه، بررسی گردد.

۶- مراجع

[1] Hindy, H. & L. Jain, "A taxonomy and survey of intrusion detection system design techniques, network threats and datasets," " International Journal of Innovative Technology and Exploring Engineering (IJITEE), 5(2), pp. 298-309, 2018.

[2] Mirjalili, S., "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," Neural Computing and Applications, 27(4), pp. 1053-1073, 2016.

[3] Yahalom, R., "Improving the effectiveness of intrusion detection systems for hierarchical data. Knowledge-Based Systems," International Journal of Innovative Technology and Exploring Engineering (IJITEE), 5(4), pp. 59-69, 2019.

[4] Farnaaz, N. & M. Jabbar, "Random forest modeling for network intrusion detection system. Procedia Computer Science," 89, pp. 213-217, 2016.

[5] Kuang, F., W. Xu, & S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection. Applied Soft Computing," 18, pp. 178-184, 2014.

[6] Faker, O. & E. Dogdu. "Intrusion detection using big data and deep learning techniques," in Proceedings of the 2019 ACM Southeast Conference, 2019.

[7] Mohammadi, S., "Cyber intrusion detection by combined feature selection algorithm," Journal of information security and applications, 44, pp. 80-88, 2019.

[8] Choudhary, S. & N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT," Procedia Computer Science, 167, pp. 1561-1573, 2020.

[9] Javaid, A., "A deep learning approach for network intrusion detection system," Eai Endorsed Transactions on Security and Safety, 3(9), p. e2, 2016.

[10] Aishwarya, C., "Intrusion Detection System using KDD Cup 99 Dataset," International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(4), pp. 3169-3171, 2020.

[11] Singh Panwar, S., Y. Raiwani, & L.S. Panwar. "Evaluation of network intrusion detection with features selection and machine learning algorithms on CICIDS-2017 dataset," in International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019, Uttaranchal University, Dehradun, India. 2019.

[12] Mafarja, M.M., "Binary dragonfly algorithm for feature selection," in 2017 International conference on new trends in computing sciences (ICTCS). 2017. IEEE.

[13] Farnaaz, N. & M. Jabbar, "Random forest modeling for network intrusion detection system," Procedia Computer Science, 89, pp. 213-217, 2016.

[14] Junker, M., R. Hoch, & A. Dengel. "On the evaluation of document analysis components by recall, precision, and accuracy," in Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318). 1999. IEEE.