مجموعه مقالات پانزدهمین کنفرانس بین‌المللی انجمن رمز ایران
دانشکده مهندسی برق، دانشگاه تربیت دبیر شهید رجایی، تهران، ۶-۷ شهریور ۱۳۹۷

یک سیستم تشخیص نفوذ چند لایه با رویکرد ترکیبی

مجید آمره‌ای[۱]، اکرم بیگی[۱]

[۱] دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران
amerehei@gmail.com, akrambeigi@sru.ac.ir

چکیده

سیستم‌های تشخیص نفوذ ابزاری مفید برای تامین امنیت شبکه های رایانه ای در مقابل نفوذگر ها هستند. این سیستم‌ها از روشهای تشخیص نفوذ مبتنی بر امضاء یا مبتنی بر ناهنجاری و یا ترکیبی از آنها استفاده می‌کنند. در سیستم‌های تشخیص نفوذ مبتنی بر امضاء از روش تطبیق داده ها با نمونه های موجود یا ایجاد قوانین استفاده می‌شود. همچنین در سیستم های تشخیص نفوذ مبتنی بر ناهنجاری میتوان از خوشه‌بندی برای شناسایی نفوذهای ناشناخته و از طبقه‌بندی برای شناسایی نفوذهای شناخته شده و تفکیک آنها از فعالیتهای نرمال استفاده کرد. در این مقاله سیستم تشخیص نفوذی با سه لایه تشخیص پیشنهاد داده ایم که از هر دو روش مبتنی بر امضاء و مبتنی بر ناهنجاری استفاده می‌کند. لایه اول با استفاده از قوانین ابتکاری برخی از نفوذها را که بسیار مشابه نمونه های نرمال هستند را تشخیص می‌دهد. لایه دوم با استفاده از خوشه‌بندی نمونه های ورودی که به مرکز خوشه ناهنجار نزدیکتر از مرکز خوشه نرمال است را به عنوان نفوذ تشخیص می‌دهد. دو مرکز خوشه توسط الگوریتم ژنتیک محاسبه می‌شود. لایه سوم نیز با استفاده از یک طبقه‌بند جنگل تصادفی نفوذها را تشخیص می‌دهد. آزمایشهای انجام شده بر روی مجموعه داده NSL-KDD و مقایسه با نتایج منتشر شده اخیری که از این مجموعه داده استفاده کرده اند، موثر بودن سیستم تشخیص نفوذ پیشنهادی را بخوبی نشان می‌دهد.

کلمات کلیدی

سیستم تشخیص نفوذ، تشخیص نفوذ مبتنی بر امضاء، تشخیص نفوذ مبتنی بر ناهنجاری، خوشه‌بندی ، طبقه‌بندی، قوانین ابتکاری.

۱- مقدمه

با گسترش استفاده از شبکه های رایانه ای در به اشتراک گذاری منابع و مطرح شدن هر چه بیشتر فضای مجازی، مقابله با تهدیدات نفوذگر ها به یکی از نیاز های ضروری در زمینه تامین امنیت اطلاعات تبدیل شده است. نفوذگرها علاوه بر روش های مختلف مانند انکار و اخلال در ارائه سرویس (DOS)[۱] ، دسترسی غیرمجاز به اطلاعات از یک ماشین راه دور (R2L)[۲] ، دسترسی غیر مجازبه دسترسی کاربر ارشد (U2R)[۳] و کاوش برای یافتن حفره‌های امنیتی (Probing) از روش‌های جدید و ناشناخته نیز بهره می‌برند. سیستم‌های تشخیص نفوذ[۴]، به عنوان یکی از مکانیزم‌های اصلی در برآوردن امنیت شبکه‌ها و سیستم‌های رایانه‌ای در کنار دیوار آتش[۵] مطرح است. این سیستم‌ها تلاش می‌کنند با بررسی ترافیک اطلاعات و رویدادها، نفوذ را تشخیص دهند.

پژوهشگرهایی که سیستم تشخیص نفوذ پیشنهاد می دهند برای نمایش کارایی آن از مجموعه داده های مختلفی استفاده می‌کنند که یکی از معروفترین آنها مجموعه داده NSL_KDD است که در سال ۲۰۰۹ توسط تاوالی و همکارانش[۳] ارائه شد. به رقم تلاش های انجام شده توسط پژوهشگران نتایج بدست آمده در این مجموعه داده با سطح مطلوب فاصله دارد. در مواجه با این مجموعه داده سیستم های تشخیص نفوذ با دو چالش مهم روبرو هستند. اولین چالش، تشخیص نفوذهایی است که بسیار مشابه نمونه‌های نرمال هستند. برخی از این نفوذها تشابه نزدیک به صد درصدی با نمونه های نرمال دارند که بعضی از حملات از نوع R2L از نمونه این حملات است[۱]. چالش دوم تشخیص حملات ناشناخته ای است که برای آنها نمونه های آموزشی وجود ندارد.

از جنبه روش تشخیص نفوذ، می‌توان سیستم های تشخیص نفوذ را به سه نوع سیستم تشخیص مبتنی بر امضاء[۶]، مبتنی بر ناهنجاری[۷] و مدل ترکیبی[۸] تقسیم بندی نمود که مدل سوم ترکیبی از دو روش اول است[۲].

بررسی پژوهش های اخیری که از مجموعه داده NSL_KDD برای ارزیابی سیستم تشخیص نفوذ خود استفاده کرده اند نشان می‌دهد که آنها به علت استفاده از تنها یک رویکرد تشخیص در حل این دو چالش به صورت همزمان موفقیت خوبی نداشته اند. ما برای حل چالش های مطرح شده در مواجهه با مجموعه داده NSL_KDD یک سیستم تشخیص نفوذ ترکیبی با سه لایه تشخیص پیشنهاد داده ایم. اگرچه استفاده از روش ترکیبی پیچیدگی زمانی را افزایش می دهد ولی در حل بهتر چالش‌های مطرح شده کمک خواهد کرد. برای حل چالش اول در لایه تشخیص اول از قوانین ابتکاری[۹] که رویکرد تشخیص مبتنی بر امضاء را دارد، استفاده کرده‌ایم. برای حل چالش دوم در لایه تشخیص دوم از روش خوشه‌بندی[۱۰] مبتنی بر الگوریتم ژنتیک[۱۱] استفاده کرده‌ایم که روش تشخیص آن مبتنی بر ناهنجاری است. برای تفکیک حملات شناخته شده از نمونه‌های عادی نیز در لایه تشخیص سوم از طبقه‌بند[۱۲] جنگل تصادفی[۱۳] استفاده کرده ایم. نتیجه آزمایش های انجام شده مقایسه آن با روش های دیگری که از مجموعه داده NSL_KDD استفاده کرده اند موثر بودن روش پیشنهادی را به خوبی نشان می‌دهد.

۲- کارهای مرتبط

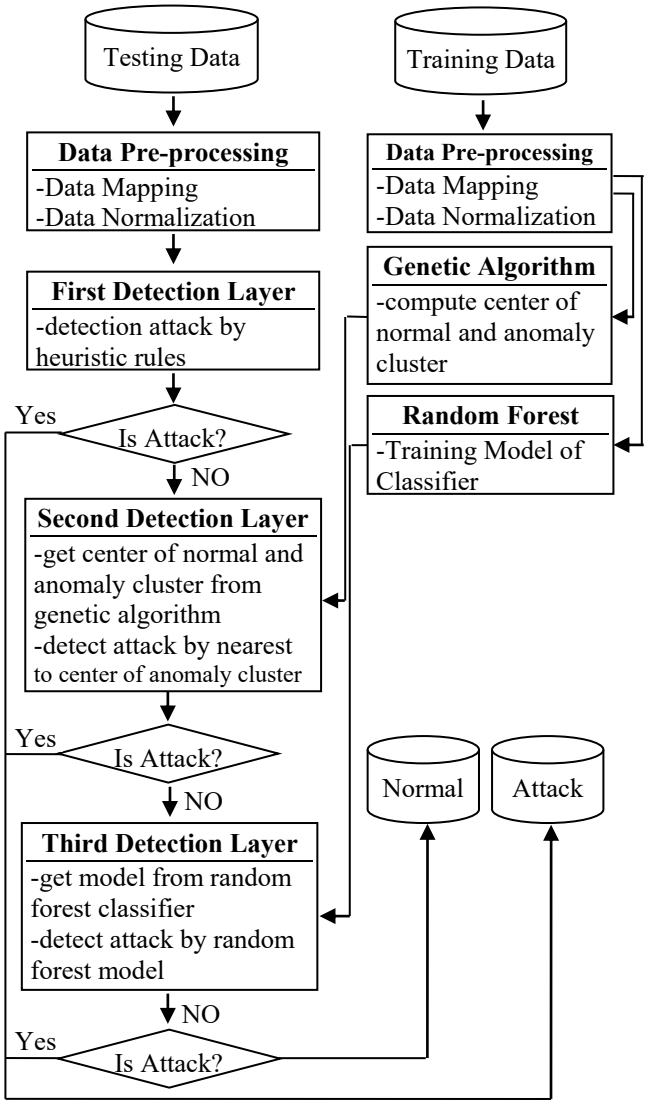
از سالی که مجموعه داده NSL_KDD ارائه شد، بسیاری از پژوهشگران برای نمایش کارایی سیستم تشخیص نفوذ خود از آن استفاده کرده اند. هر یک از آنها با الگوریتم های متفاوت سعی در بهبود کارآیی سیستم خود در مواجهه با این مجموعه داده داشته اند که در ادامه دو مورد از آنها را که در تحقیقات اخیر نسبت به سایر روش های ارائه شده نتایج بهتر و موفق تری داشته اند را مطرح می‌کنیم.

ین و همکارانش[۴] یک سیستم تشخیص نفوذ مبتنی بر یادگیری عمیق با استفاده از شبکه های عصبی مکرّر[۱۴] (RNN) پیشنهاد دادند که در مقایسه با روش های یادگیری سنتی نتایج بهتری داشت. این روش یادگیری برای آموزش به یک مجموعه داده آموزشی برچسب دار نیاز دارد و به همین دلیل در تشخیص نفوذهای ناشناخته ناتوان است. اشفق و همکارانش[۵] یک روش یادگیری نیمه نظارتی مبتنی بر فازی را پیشنهاد دادند. در این روش ابتدا یک مجموعه داده آموزشی بدون برچسب توسط یک طبقه‌بند شبکه عصبی[۱۵] به سه گروه فازی کم، متوسط و زیاد خوشه‌بندی می‌شود و سپس گروه های فازی کم و زیاد با یک مجموعه داده آموزشی برچسب دار ادغام می‌گردد. استفاده از مجموعه داده آموزشی بدست آمده باعث بهبود کارایی طبقه‌بندی سیستم تشخیص نفوذ می‌شود. اگر چه این روش در تشخیص نفوذهای ناشناخته توانست مناسب عمل کند ولی در تشخیص حملاتی که بسیار مشابه نمونه های نرمال است موفقیت چشمگیری نداشت.

در این مقاله بهبود نتایج روش پیشنهادی را نسبت به تحقیقات اخیر نشان خواهیم داد.

۳- الگوریتم پیشنهادی

در این بخش الگوریتم پیشنهادی خود را برای طراحی سیستم تشخیص نفوذ شرح می دهیم. این الگوریتم شامل دو فرایند اصلی است. ابتدا فرایند آموزش خوشه‌بند و طبقه‌بند با استفاده از مجموعه داده آموزشی انجام شده و سپس فرایند تشخیص و تفکیک حملات از نمونه های نرمال مجموعه داده آموزشی انجام می‌شود. روش کار سیستم تشخیص نفوذ پیشنهادی در شکل (۱) نمایش داده شده است.



شکل (۱) : سیستم تشخیص نفوذ پیشنهادی

روش کار الگوریتم به این صورت است که ابتدا عملیات نگاشت و پیش پردازش بر روی مجموعه داده آموزشی و آزمایشی مطابق روش پیشنهاد شده در [۶] انجام می‌شود. سپس تمامی داده های مجموعه داده آموزشی به الگوریتم خوشه‌بندی مبتنی بر ژنتیک و الگوریتم طبقه‌بندی جنگل تصادفی ارسال می‌شود. الگوریتم خوشه‌بند مبتنی بر ژنتیک، داده ها را به دو خوشه ناهنجار و نرمال تقسیم کرده و مراکز این خوشه ها را برای لایه تشخیص دوم ارسال می‌کند. الگوریتم طبقه‌بندی جنگل تصادفی نیز با استفاده از این داده‌ها آموزش داده شده و مدل بدست آمده از آن به لایه تشخیص سوم ارسال می‌شود. در مرحله بعد، هر یک از نمونه های موجود در مجموعه داده آزمایشی برای برچسب زدن نوبت به نوبت انتخاب شده و به ترتیب وارد سه لایه تشخیص می‌شود. این سه لایه بترتیب از روش تشخیص مبتنی بر قوانین ابتکاری، روش تشخیص مبتنی بر خوشه‌بندی ژنتیک و روش تشخیص مبتنی

بر طبقه‌بند جنگل تصادفی استفاده می‌کنند. هر یک از این لایه ها نمونه های آزمایشی را بررسی کرده و در صورت تشخیص ناهنجاری بر روی آن برچسب حمله می زنند. در صورتی که هیچ یک از این لایه ها نمونه آزمایشی را ناهنجار تشخیص نداد برچسب نرمال بر روی نمونه زده می‌شود. این روند تا برچسب زدن تمام نمونه های موجود در مجموعه داده آزمایشی ادامه می یابد. در ادامه به جزییات هر یک از سه لایه تشخیص می‌پردازیم.

۳-۱- لایه تشخیص اول مبتنی بر قوانین ابتکاری

بسیاری از حملات از نوع R2L شباهت زیادی با نمونه‌های نرمال دارد[۱] و طبقه‌بند ها در تشخیص آنها به خوبی عمل نمی‌کنند. ما برای حل این چالش از یک لایه تشخیص مبتنی بر قوانین ابتکاری استفاده کرده ایم. یک قانون ابتکاری در سیستم تشخیص نفوذ، دانش یک انسان خبره در مورد امضاء حملات به صورت یک فرمول موثر در قالب "اگر، آنگاه" است[۱]. یک قانون ابتکاری را می‌توان با بررسی رفتار و امضاء یک حمله بدست آورد.

نمونه ای از قوانین ابتکاری را می‌توان در حملات quess_password و warezmaster که زیر مجموعه حملات از نوع R2L است، بکار برد. در حملات از نوع quess_password نفوذ کننده کلمات عبور مختلف را بارها و بارها آزمایش می‌کند تا بتواند یک کلمه عبور مناسب برای دسترسی به اطلاعات پیدا کند. با بررسی اینکه کاربر در هر رویداد شبکه ای چند بار در ورود به سیستم با شکست مواجعه شده و در نهایت آیا موفق به ورود شده یا خیر، می‌توان این نوع از حملات را تا حد قابل توجهی تشخیص داد. با توجه به اینکه در مجموعه داده NSL_KDD تعداد شکست در ورود به سیستم در ویژگی num_failed_logins و وضعیت ورود به سیستم در ویژگی logged_in ثبت شده است ما می‌توانیم یک قانون ابتکاری با تمرکز بر این دو ویژگی بنویسیم. در حملات از نوع warezmaster نیز می‌توان از قانون ابتکاری ارائه شده در [۱] استفاده کرد. الگوریتم این لایه تشخیص را می‌توانید در شکل (۲) ببینید.

Algorithm : First Detection Layer - heuristic rules

Input:

- Ts_i : Instance Data of Testing Data Set

Output:

- isattack : 0 is not attack and 1 is attack

```
1:  //rule 1: guess_passwd detection
2:  if (num_failed_logins >0) and (is_guest_login==0)
3:        Isattack=1;
4:        Return;
5:  //rule 2: warezmaster detection
6:  if ((protocol_type == tcp) and (service ==ftp
    or service ==ftp_data ))
7:        if ((duration > 0 and  src_bytes >0 and dst_bytes ==0)
          or (hot >0 and is_guest_login ==1))
8:                Isattack=1;
9:                Return;
```

شکل (۲) : الگوریتم لایه تشخیص اول مبتنی بر قوانین ابتکاری

این لایه تشخیص شامل دو قانون ابتکاری است. در قانون ابتکاری اول اگر تعداد شکست در ورود به سیستم بیشتر از یک بار بوده و نیز کاربر در ورود به سیستم ناموفق باشد، نمونه ورودی حمله تشخیص داده می‌شود. در قانون ابتکاری دوم نیز ابتدا نوع پروتکل و سرویس استفاده شده بررسی می‌شود. اگر پروتکل از نوع TCP و سرویس از نوع FTP یا FTP_DATA باشد آنگاه در صورت تحقق یکی از دو شرط زیر نمونه ورودی حمله تشخیص داده می‌شود:

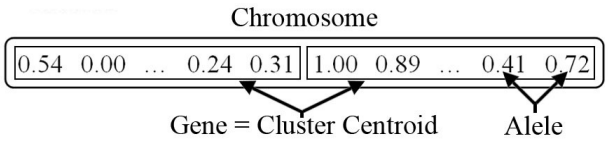
- مدت اتصال و میزان بایت ارسال شده بیشتر از صفر و میزان بایت دریافت شده صفر باشد.
- کاربر مهمان یک یا چند پوشه یا فایل در منبع ایجاد کرده باشد.

۳-۲- لایه تشخیص دوم مبتنی بر خوشه‌بندی

بخشی از حملاتی که به منابع سیستم ها انجام می‌شود ناشناخته بوده و برای آنها نمونه های آموزشی وجود ندارد. به همین دلیل استفاده از یک طبقه‌بند کمکی به تشخیص این نوع از حملات نمی‌کند. برای حل این چالش می‌توان از یک روش خوشه‌بندی برای تشخیص حملات ناشناخته استفاده کرد. الگوریتم های خوشه‌بندی مختلفی مانند الگوریتم کامینز و الگوریتم ژنتیک برای خوشه‌بندی حملات موجود است. با توجه به نتایج گزارش شده در [۷] ما در سیستم تشخیص نفوذ پیشنهادی از الگوریتم ژنتیک برای خوشه‌بندی استفاده کرده ایم.

الگوریتم ژنتیک یک تکنیک بهینه سازی تطبیقی و فرا ابتکاری است. این الگوریتم از اصل داروین که مبتنی بر بقای بهترین ها است الهام گرفته شده است [۸]. این الگوریتم مبتنی بر رویکردی است که همه افراد در یک نسل برای بدست آوردن منابع با یکدیگر رقابت می‌کنند و موفق ترین افراد اجازه تولید فرزند را خواهند داشت. تکرار این رویکرد موجب بهبود هر نسل نسبت به نسل قبل می‌شود.

به طور کلی، الگوریتم ژنتیک یک روش جستجوی مبتنی بر جمعیت است که هر کدام از آنها به عنوان یک کروموزم[۱۶] با طول ثابت به صورت دودویی بازنمایی می شوند. جدا از این، کروموزوم ها می‌توانند به صورت مقادیر درختی، جایگشتی و حقیقی نیز بازنمایی شوند[۷]. ما از مقادیر حقیقی برای بازنمایی استفاده کرده‌ایم. هر کروموزوم شامل ۲ ژن[۱۷] و هر ژن دارای ۴۱ آلل[۱۸] است. ژن ها در الگوریتم ما نشان دهنده مراکز خوشه ها و آلل ها نشان دهنده ویژگی های هر مرکز هستند. جمعیت اولیه به صورت تصادفی از مجموعه داده انتخاب می‌شود. ساختار پیشنهادی یک کروموزم در شکل (۳) نشان داده شده است.



شکل (۳) : ساختار یک کروموزوم

پس از تولید کروموزم ها، کیفیت هر یک از آنها با استفاده از یک تابع شایستگی[۱۹] ارزیابی می‌شود تا مناسب ترین کروموزم ها برای تولید فرزندان نسل بعدی انتخاب شوند. الگوریتم تابع شایستگی پیشنهادی در شکل (۴) نمایش داده شده است.

Algorithm : fitness of genetic clustering

Input:
- ch:chromosome;
- Tr_d : Data of Training Data Set

Output:
- F_v: Value of Fitness

```
1: set label for all instance of tr_d by nearest to Gen1 or Gen2;
2: split tr_d to claster1 and claster2 by labels;
3: compute intra and extra quality of classter1 and claster2;
4: set Fitness value(F_v) for qualities;
5: Return F_v;
```

شکل (۴) : الگوریتم تابع شایستگی

در این تابع شایستگی، خوشه هر یک از نمونه های مجموعه داده آموزشی بر اساس نزدیکی فاصله به هر یک از مراکز پیشنهاد شده تعیین می‌شود. پس از تکمیل خوشه‌بندی، فاصله درون خوشه ای و بین خوشه ای هر دو خوشه به صورت زیر محاسبه می‌شود:

- فاصله درون خوشه ای: در هر یک از این خوشه ها، مجموع فاصله همه نمونه ها با مرکز آن خوشه محاسبه می‌شود. این جمع نشان‌دهنده فاصله درون خوشه‌ای نمونه ها است.
- فاصله بین خوشه ای: میانگین فاصله دو نقطه مرکزی خوشه ها نسبت به یکدیگر محاسبه می‌شود. سپس مجموع فاصله نمونه‌های هر خوشه با میانگین فاصله مراکز خوشه‌ها محاسبه می‌شود. این جمع نشان‌دهنده فاصله بین خوشه‌ای نمونه ها است.

هرچه فاصله درون خوشه‌ای کمتر و فاصله بین خوشه‌ای بیشتر باشد، تابع برازندگی مقدار بیشتری را به‌عنوان خروجی نمایش می‌دهد.

پس از تولید جمعیت اولیه کروموزم ها و تعیین تابع شایستگی برای هر یک از آنها، مراحل انتخاب والدین، انجام عمل بازترکیب و جهش و انتخاب بازماندگان و نیز مقدار دهی پارامتر ها لازم طبق روش پیشنهادی[۷] انجام می‌شود.

پس از پایان کار الگوریتم ژنتیک خوشه‌بندی، ۲ مرکز خوشه بدست می‌آید که یکی مرکز خوشه نرمال و دیگری مرکز خوشه ناهنجار است. در ادامه کار این دو مرکز خوشه برای لایه تشخیص دوم ارسال می‌شود. این لایه تشخیص با استفاده از این مراکز خوشه ها نمونه های آزمایشی ورودی را برچسب می زند. شکل (۵) الگوریتم لایه تشخیص دوم را نمایش می‌دهد.

Algorithm : Second Detection Layer - clustering

Input:
- Ts_i : Instance Data of Testing Data Set
- Ts_c1 : center of clusters anomaly
- Ts_c2 : center of clusters normal

Output:
- isattack : 0 is not attack and 1 is attack

```
1: if (distacnce(Ts_i ,Ts_c1) < distacnce(Ts_i ,Ts_c2) )
2:         isattack=1;
3:         Return;
```

شکل (۵) : الگوریتم لایه تشخیص دوم مبتنی بر خوشه‌بندی

روش کار این لایه تشخیص به این صورت است که اگر نمونه آزمایشی ورودی به مرکز خوشه ناهنجار نزدیک تر از مرکز خوشه نرمال بود بر آن برچسب ناهنجار زده می‌شود.

۳-۳- لایه تشخیص سوم مبتنی بر جنگل تصادفی

برخی از حملات در لایه های تشخیص اول و دوم که به ترتیب از روش قوانین ابتکاری و روش خوشه‌بندی استفاده می‌کنند تشخیص داده نمی شوند. ما برای افزایش دقت سیستم تشخیص نفوذ در لایه سوم تشخیص از یک طبقه‌بند جنگل تصادفی برای بررسی نهایی نمونه ها استفاده کرده‌ایم.

جنگل تصادفی یک طبقه‌بند مجموعه ای است که از چند طبقه‌بند درخت تصمیم[۲۰] تشکیل شده است. هر یک از درخت ها برای هر نمونه ورودی یک پیش بینی ارائه می‌دهد. در نهایت کلاسی که بیشترین تعداد رای را دارد به عنوان برچسب ورودی انتخاب می‌شود. این فرآیند را جنگل تصادفی می‌نامند. الگوریتم جنگل تصادفی می‌تواند دقت پیش بینی را نسبت به درخت طبقه‌بند فردی افزایش دهد. در درخت فردی با تغییرات کوچک در مجموعه آموزشی، بی ثباتی به وجود می‌آید که باعث اختلال در دقت پیش بینی در نمونه آزمایشی می‌شود. اما گروهی بودن الگوریتم جنگل تصادفی باعث سازگاری با تغییرات می‌شود و بی ثباتی را از بین می برد[۹]. طبقه‌بند جنگل تصادفی دو مرحله آموزش و آزمایش دارد. در سیستم تشخیص پیشنهادی ابتدا طبقه‌بند جنگل تصادفی با استفاده از مجموعه داده آموزشی، آموزش داده شده و سپس مدل بدست آمده به لایه تشخیص سوم که از مدل آموزش دیده طبقه‌بند استفاده می‌کند ارسال می‌شود. شکل (۶) الگوریتم لایه تشخیص سوم مبتنی بر طبقه‌بند جنگل تصادفی را نمایش می‌دهد.

لایه تشخیص سوم با استفاده از طبقه‌بند جنگل تصادفی که آموزش دیده است نمونه های آزمایشی ورودی بررسی کرده و در صورت تشخیص نفوذ آن را اعلام می‌کند.

Algorithm : Third Detection Layer - Classifier

Input:
- Ts_i : Instance Data of Testing Data Set
- Tr_structure : structure of random forest

Output:
- isattack : 0 is not attack and 1 is attack

```
1: if (Tr_structure (Ts_i) == attack)
2:         Isattack=1;
3:         Return;
```

شکل (۶) : الگوریتم لایه تشخیص سوم مبتنی بر طبقه‌بندی

۴- آزمایش ها و نتایج

۴-۱- مجموعه داده

اولین رویداد سیستم تشخیص نفوذ در سال ۱۹۹۸ با حمایت سازمان دارپا[۲۱] انجام ش [۱۰]. در این رویداد ، یک سناریوی حمله سایبری به پایگاه نیروی هوایی شبیه سازی شد. آنها یک سال بعد، در سال ۱۹۹۹، با بهبود نظریه شان به وسیله انجمن امنیت کامپیوتر، رویداد مشابهی را تکرار کردند [۱۱]. در هفت هفته، داده های خام TCP/IP شبکه جمع آوری شد. این داده ها خام بود

ولی پژوهشگران نیاز به استخراج ویژگی از آن برای استفاده آن در الگوریتم های یادگیری داشتند. محققان دیگری [۱۰] با ارائه یک روش استخراج ویژگی برای این مجموعه داده خام، موفق به پیروزی در رقابت بین المللی کشف دانش و استخراج داده KDD شدند و مجموعه داده آنها به نام KDDCUP99 مطرح شد. این مجموعه داده محک بسیار محبوبی در زمینه تشخیص نفوذ است[۵].

در [۳] تعدادی از کاستی های مجموعه داده KDDCUP99 که در عملکرد ارزیابی سیستم بشدت تاثیر گذار است و به صورت آماری کشف شده، معرفی شده اند. همچنین در این تحقیق یک مجموعه داده بهبود یافته که NSL-KDD نامیده می شد، برای مقایسه مدل های تشخیص نفوذ مختلف، پیشنهاد شده است.

مجموعه داده NSL-KDD شامل ۴۱ ویژگی همانند مجموعه داده KDDCUP99 است و برچسب کلاس ها نیز مشابه است. تعداد از ویژگی‌ها مقادیر پیوسته و تعدادی از آنها غیر پیوسته هستند. به علت کارآیی بهتر سیستم تشخیص نفوذ پیشنهادی ابتدا مقادیر غیر پیوسته به مقادیر پیوسته عددی نگاشت شده و سپس تمامی ۴۱ ویژگی های بین ۰ تا ۱ نرمال می‌شوند. در این مقاله از روش نگاشت و نرمال سازی ویژگی ها که در [۶] بیان شده، بهره گرفته ایم.

۴-۲- معیارهای ارزیابی

در فرآیند تشخیص نفوذ، چهار حالت تشخیص برای یک رویداد رخ می‌دهد که عبارتند از:

- مثبت صحیح[۲۲] (TP) : نمونه ورودی در پایان فرآیند تشخیص به طور صحیح به عنوان نفوذ تشخیص داده شود.
- منفی صحیح[۲۳] (TN) : نمونه ورودی در پایان فرآیند تشخیص به طور صحیح به عنوان نرمال تشخیص داده شود.
- مثبت غلط[۲۴] (FP) : نمونه ورودی از منظر امنیتی بی خطر است اما در پایان فرآیند تشخیص به عنوان نفوذ تشخیص داده شود.
- منفی غلط[۲۵] (FN) : نمونه ورودی از منظر امنیتی خطرناک است ولی در پایان فرآیند تشخیص به عنوان نرمال تشخیص داده شود.

معیار های مهم ارزیابی سیستم های تشخیص نفوذ با توجه به این چهار حالت تشخیص به صورت زیر محاسبه می شوند:

- دقت کل[۲۶] (AC) : درصد تعداد نمونه هایی که بدرستی برچسب زده شده در مقایسه با کل نمونه های موجود، همانطور که در فرمول (۱) نشان داده شده است.

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \quad (۱)$$

- نرخ تشخیص[۲۷] (DR) : نشان دهنده درصد تعداد نمونه هایی که بدرستی شناسایی شده بر تعداد کل نمونه های ناهنجار، همانطور که در فرمول (۲) نشان داده شده است.

$$DR = \frac{TP}{TP + FN} \quad (۲)$$

- نرخ مثبت غلط[۲۸] (FPR) : درصد تعداد نمونه های رد شده نادرست تقسیم بر کل نمونه های نرمال، همانطور که در فرمول (۳) نشان داده شده است.

$$FPR = \frac{FP}{FP + TN} \quad (۳)$$

با توجه به این معیارها هر چه مقدار دقت کل و نرخ تشخیص بالاتر و نرخ مثبت غلط کمتر باشد، کارایی سیستم تشخیص نفوذ بهتر خواهد بود[۴].

۴-۳- نتایج آزمایش ها

در این بخش نتایج آزمایش های انجام شده بر روی سیستم تشخیص نفوذ پیشنهادی نشان خواهیم داد. در این آزمایش ها از دو مجموعه داده Kddtest+ و Kddtest-21 برای ارزیابی استفاده شده است. مهمترین تفاوت این دو مجموعه آموزشی تعداد نمونه های نرمال است. تعداد نمونه های نرمال در مجموعه داده آموزشی KDDtest+ به تعداد ۷۵۵۹ نمونه از مجموعه داده آموزشی KDDtest-21 بیشتر است.

جدول (۱) نتیجه انجام آزمایش بر روی این دو مجموعه داده با استفاده از سیستم تشخیص پیشنهاد شده را نشان می‌دهد.

جدول (۱) : نتایج آزمایش ها

Dataset	TP	TN	FP	FN
Kddtest+	۱۰۲۷۳	۹۲۸۳	۴۲۸	۲۵۶۰
Kddtest-21	۷۱۳۸	۱۷۴۹	۴۰۳	۲۵۶۰

مقادیر معیار های ارزیابی با توجه به نتایج آزمایش ها در جدول (۲) نمایش داده شده است.

جدول (۲) : مقادیر معیارهای ارزیابی

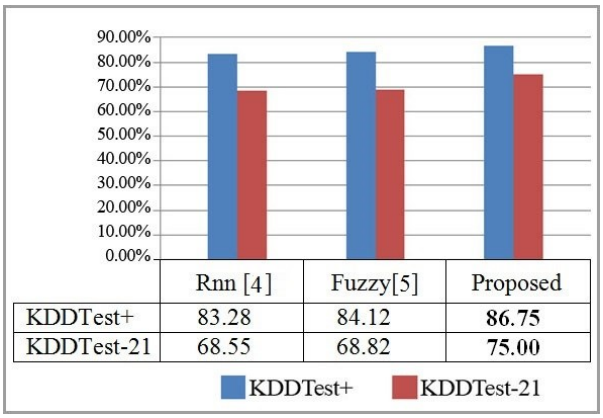
Dataset	FPR	DR	AC
Kddtest+	۴/۴۰	۸۰/۰۵	۸۶/۷۵
Kddtest-21	۱۸/۷۲	۷۳/۶۰	۷۵/۰۰

در ادامه نتایج بدست آمده الگوریتم پیشنهادی را با الگوریتم پیشنهاد شده اشفق و همکارانش [۵] و الگوریتم ین و همکارانش [۴] که بهترین نتایج را در مجموعه داده NSL-KDD دارند مقایسه می‌کنیم. ین و همکارانش مقادیر دقت کل و نرخ تشخیص و نرخ مثبت غلط را برای مجموعه داده Kddtest+ بترتیب ۸۳,۲۸ ، ۷۲,۹۵ و ۳,۰۶ گزارش کرده اند که مقایسه مقادیر نشان می‌دهد بغیر از معیار نرخ مثبت غلط در دو معیار دیگر نتایج سیستم تشخیص نفوذ پیشنهادی بهتر بوده است. اشفق و همکارانش گزارشی درباره نرخ تشخیص و نرخ مثبت غلط ارائه نکرده اند و به ارائه دقت کل اکتفا نموده‌اند.

در شکل (۷) دقت کل الگوریتم پیشنهادی با دقت کل دو روش دیگر مقایسه شده است که نشان دهنده موفقیت آمیز بودن روش پیشنهادی است.

۵- نتیجه گیری

ما در این مقاله یک سیستم تشخیص نفوذ با رویکرد ترکیبی پیشنهاد داده ایم. این سیستم دارای سه لایه تشخیص است که هر یک با روشی تشخیص خاص خود سعی در تفکیک حملات از نمونه های نرمال دارد. لایه تشخیص اول مبتنی بر تشخیص امضاء است و بر اساس قوانین ابتکاری کار تشخیص را انجام می‌دهد. لایه تشخیص دوم و سوم مبتنی بر ناهنجاری هستند که لایه دوم بر اساس یک روش خوشه‌بندی و لایه سوم بر اساس یک روش طبقه‌بندی کار تشخیص را انجام می‌دهد. آزمایش های انجام شده عملکرد بهتر این روش را نسبت به روش های مطرح شده اخیر که از مجموعه داده



شکل (۷) : مقایسه دقت کل الگوریتم ها

NSL-KDD برای ارزیابی سیستم تشخیص نفوذشان استفاده کرده اند را نشان می‌دهد. به علت اینکه فضای مجازی دارای محیط وسیع بوده و حجم ترافیک اطلاعات در آن بسیار زیاد است و نیز استفاده از روش های ترکیبی باعث کاهش سرعت تشخیص می شود، ما در کارهای آینده در نظر داریم از یک رویکرد چند عامله برای بهبود سرعت پردازش روش خود استفاده کنیم.

مراجع

[1] Sabhnani, Maheshkumar, and Gürsel Serpen. "KDD Feature Set Complaint Heuristic Rules for R2L Attack Detection." Security and Management. 2003.

[2] Fuchsberger, Andreas. "Intrusion detection systems and intrusion prevention systems." Information Security Technical Report 10.3 (2005): 134-139.

[3] Tavallaee, Mahbod, et al. "A detailed analysis of the KDD CUP 99 data set." Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on. IEEE, 2009.

[4] Yin, Chuanlong, et al. "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks." IEEE Access 5 (2017): 21954-21961.

[5] Ashfaq, Rana Aamir Raza, et al. "Fuzziness based semi-supervised learning approach for intrusion detection system." Information Sciences 378 (2017): 484-497.

[6] Al-Yaseen, Wathiq Laftah, Zulaiha Ali Othman, and Mohd Zakree Ahmad Nazri. "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system." Expert Systems with Applications 67 (2017): 296-303.

[7] Aissa, Naila Belhadj, and Mohamed Guerroumi. "A genetic clustering technique for Anomaly-based Intrusion Detection Systems." Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on. IEEE, 2015.

[8] [7] Davis, L. (1991). Handbook of genetic algorithms. New York: Van Nostrand Reinhold, (n.d.).

[9] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[10] Cunningham, Robert K., et al. Evaluating intrusion detection systems without attacking your friends: The 1998 DARPA intrusion detection evaluation. MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, 1999.

[11] Lippmann, Richard P., et al. "Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation." DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings. Vol. 2. IEEE, 2000.

زیرنویس‌ها

[1] Denial Of Service
[2] Remote To Local
[3] User To Root
[4] Intrusion Detection Systems
[5] Firewall
[6] Signature-Base Detection
[7] Anomaly-Base Detection
[8] Hybrid
[9] Heuristic Rules
[10] Clustering
[11] Genetic Algorithm
[12] Classification
[13] Random Forest
[□□] Recurrent Neural Networks
[□□] Neural Networks
[16] Chromosome
[17] Gen
[18] Allele
[19] Fitness Function
[20] Decision Trees
[21] Darpa
[22] True Positive
[23] True Negative
[24] False Positive
[25] False Negative
[26] Accuracy
[27] Detection Rate
[28] False Positive Rate