4th International Conference on Web Research
April 25-26, 2018; Tehran, IRAN
ICWR2018
کنفرانس بین المللی وب پژوهی
چهارمین
۵ و ۶ اردیبهشت ۱۳۹۷؛ تهران، ایران

Using Genetic Algorithm to Improve Random Forest Algorithm in Order to Detect DDoS Attacks in Cloud Computing Platform

[1]Ali Mahmodi Derakhsh, [2]Parisa Daneshjoo, [3]Changiz Delara
[1] Department of Computer Engineering, West Tehran Branch, Islamic Azad University, Tehran, Iran
Mahmodi.Ali@wtiau.ac.ir
[2] Department of Computer Engineering, West Tehran Branch, Islamic Azad University, Tehran, Iran
Daneshjoo.p@wtiau.ac.ir
[3] Department of Computer Engineering, West Tehran Branch, Islamic Azad University, Tehran, Iran
Delara.c@wtiau.ac.ir

Abstract

Devices such as routers, switches or firewalls are the most vital connections in communication network among physical machines in a cloud computing environment. In the absence of security on the network, intruders are allowed to access the equipment and configure it in the way they want to. Hence, a method suggested to deal with denial-of-service (DoS) attacks in the cloud computing platform is one of the essential and most important security issues in this area. This study tends to provide a smart method based on random forest algorithm focusing on genetic algorithms for detecting DoS attacks. Through different network streams, network streams which trigger DoS and DDoS attacks are very important.

The main idea of this study is to use random forest algorithm to identify DoS attacks, which is the main reason for optimizing this algorithm using genetic algorithms. In this method, an optimal subset of the set of features is extracted using genetic algorithms, and this optimal subset is used for random forest learning. Results of the experiments carried out and comparison of the suggested method with other methods indicate proper accuracy and operation of the suggested method.

Keywords: Cloud computing, Network security, Denial-of-service attacks, Genetic algorithms, Random forest

4th International Conference on Web Research
April 25-26, 2018; Tehran, IRAN
ICWR2018
کنفرانس بین المللی وب پژوهی
چهارمین
۵ و ۶ اردیبهشت ۱۳۹۷؛ تهران، ایران

بکارگیری الگوریتم ژنتیک برای بهبود الگوریتم جنگل تصادفی به منظور تشخیص حملات محروم سازی از سرویس توزیع شده در بستر رایانش ابری

علی محمودی درخش[۱]، پریسا دانشجو[۲]، چنگیز دل آرا[۳]

[۱] دانشجوی کارشناسی ارشد، گروه کامپیوتر، واحد تهران غرب، دانشگاه آزاد اسلامی ، تهران
Mahmodi.Ali@wtiau.ac.ir
[۲] استادیار و عضو هیئت علمی، گروه کامپیوتر، واحد تهران غرب، دانشگاه آزاد اسلامی ، تهران
Daneshjoo.p@wtiau.ac.ir
[۳] استادیار و عضو هیئت علمی، گروه کامپیوتر، واحد تهران غرب، دانشگاه آزاد اسلامی ، تهران
Delara.c@wtiau.ac.ir

چکیده

تجهیزاتی همچون مسیریاب، سوئیچ یا دیوارهای آتش به‌عنوان حیاتی‌ترین اتصالات در شبکه ارتباطی میان ماشین‌های فیزیکی در یک محیط رایانش ابری هستند. در صورت عدم وجود امنیت در شبکه، به‌نفوذگران به شبکه اجازه داده می‌شود که با دستیابی به تجهیزات، امکان پیکربندی آن‌ها را به‌گونه‌ای که تمایل دارند عمل کنند. ازاین‌رو ارائه روشی برای مقابله با حملات محروم‌سازی از سرویس[۱] در بستر رایانش ابری یکی از ضروریات و مهم‌ترین مسائل امنیتی این حوزه می‌باشد. در این تحقیق سعی شده روشی هوشمند و مبتنی بر الگوریتم درخت تصادفی[۲] با تمرکز بر روی الگوریتم ژنتیک[۳] برای شناسایی حملات محروم‌سازی از سرویس ارائه شود. از میان جریان‌های شبکه‌ای متفاوت، جریان‌های شبکه‌ای که موجب بروز حملات DoS و DDoS [۴] می‌شوند از اهمیت بسیار بالایی برخوردارند.

ایده اصلی در این تحقیق استفاده از روش‌های داده‌کاوی و الگوریتم درخت تصادفی برای شناسایی حملات محروم‌سازی از سرویس است که نکته اصلی در بهینه‌سازی این الگوریتم با استفاده از الگوریتم‌های ژنتیک می‌باشد. طی این روش زیرمجموعه بهینه از مجموعه ویژگی‌های با استفاده از الگوریتم‌های ژنتیک استخراج می‌شود و این زیرمجموعه بهینه برای یادگیری مدل درخت تصادفی مورد استفاده قرار می‌گیرد. نتایج آزمایش‌های انجام‌شده و مقایسه آن با روش معمول حاکی از دقت و عملکرد مناسب روش ارائه‌شده دارد.

کلمات کلیدی

رایانش ابری، امنیت شبکه، حملات محروم سازی از سرویس، الگوریتم‌های ژنتیک، جنگل تصادفی

۱- مقدمه

برای تأمین امنیت بر روی یک شبکه، یکی از بحرانی‌ترین و خطیرترین مراحل، تأمین امنیت دسترسی و کنترل تجهیزات شبکه است. تجهیزاتی همچون مسیریاب، سوئیچ یا دیوارهای آتش که به‌عنوان حیاتی‌ترین اتصالات در شبکه ارتباطی میان ماشین‌های فیزیکی در یک محیط رایانش ابری هستند. موضوع امنیت تجهیزات به دو علت اهمیت ویژه‌ای می‌یابد[1]عدم وجود امنیت تجهیزات در شبکه، به‌نفوذگران به شبکه اجازه می‌دهد که با دستیابی به تجهیزات، امکان پیکربندی آن‌ها را به‌گونه‌ای که تمایل دارند عمل کنند، از این طریق هرگونه نفوذ و سرقت اطلاعات و یا هر نوع صدمه دیگری به شبکه محیط رایانش ابری، توسط نفوذگر، امکان‌پذیر خواهد شد. برای جلوگیری از خطرهای محروم‌سازی از سرویس، تأمین امنیت تجهیزات بر روی شبکه الزامی است. توسط این حمله‌ها نفوذگران می‌توانند سرویس‌هایی را در شبکه از کار بیاندازند. ازاین‌رو ارائه روشی برای مقابله با حملات محروم‌سازی از سرویس در بستر شبکه محیط رایانش ابری یکی از ضروریات و مهم‌ترین مسائل امنیتی این حوزه می‌باشد. ازاین‌رو در این تحقیق سعی بر آن داریم تا روش هوشمند و مبتنی بر الگوریتم درخت تصمیم با

تمرکز بر روی الگوریتم‌های ژنتیک برای شناسایی حملات محروم‌سازی از سرویس ارائه دهیم. ایده اصلی در این تحقیق استفاده از الگوریتم درخت تصمیم برای شناسایی حملات محروم‌سازی از سرویس است که نکته اصلی در بهینه‌سازی این الگوریتم‌ها با استفاده از الگوریتم‌های ژنتیک می‌باشد که روشی کاملاً جدید و نو می‌باشد.

۱-۱- کارهای پیشین

در[2] به بررسی روش‌های مختلف یادگیری ماشین و داده‌کاوی که در شناسایی حملات محروم‌سازی از سرویس از آن‌ها استفاده‌شده است، می‌پردازد. روش‌هایی مانند ماشین بردار پشتیبان، شبکه‌های بیزین و درخت تصمیم ازجمله مهم‌ترین الگوریتم‌هایی است که در این مقاله مروری مورد ارزیابی قرار گرفته است. این مقاله برای آشنایی با روش‌های مقابله با حملات محروم‌سازی از سرویس که از فن‌های داده‌کاوی استفاده می‌کنند، منبعی جامع، کامل و جدید هست.

در [3] به شناسایی حملات محروم‌سازی از سرویس با استفاده از یک طبقه‌بند پرداخته شده است. طبقه‌بند مورد استفاده در این روش ماشین بردار پشتیبان است که در نوع خود یکی از کاراترین طبقه‌بندهای ممکن هست. نکته اصلی در این مقاله استفاده از این طبقه‌بند برای شناسایی حملات محروم‌سازی از سرویس در محیط شبکه مجهز به SDN[5] می‌باشد.

در[4] به ارائه چارچوب و چارچوبی برای شناسایی حملات محروم‌سازی از سرویس با استفاده از الگوریتم‌های ژنتیک پرداخته است. با توجه به الگوریتم‌های ژنتیک برای حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرند، در این مقاله شناسایی حملات محروم‌سازی از سرویس به یک مسئله بهینه‌سازی نگاشت شده است و از الگوریتم‌های ژنتیک برای حل آن استفاده می‌شود. آزمایش‌ها نشان می‌دهد که روش ارائه شده از دقت مناسبی برخوردار است.

در [5] از روش انتخاب ویژگی برای یافتن ویژگی‌های بهینه برای تشخیص نفوذ و شناسایی حملات DDoS استفاده شده است. ویژگی اصلی این روش در استفاده از فیلترها برای انتخاب مجموعه ویژگی بهینه است. درنهایت نیز از الگوریتم ماشین بردار پشتیبان برای تشخیص نفوذ استفاده شده است. نتایج حاکی از دقت بسیار مناسب این روش می‌باشد.

در [6] از فیلترها برای کاهش ویژگی و شناسایی مجموعه ویژگی بهینه استفاده‌شده است. تفاوت این روش با پاراگراف قبل در این است که در این روش اولاً از فیلترهای چندگانه استفاده شده است و دوما روش ارائه‌شده یک روش بلادرنگ است و قابلیت اجرا بر روی داده‌های جریانی را دارد. این روش تمرکز خود را بر روی تشخیص نفوذ در محیط رایانش ابری قرار داده است.

در [7] نیز تمرکز اصلی بر روی انتخاب و شناسایی ویژگی است. برای این منظور از روش آنتروپی فازی استفاده شده است. علاوه بر این برای بهینه‌سازی و انتخاب ویژگی‌های بهینه نیز از روش کلونی مورچگان استفاده شده است. ویژگی مهم دیگری که این روش دارد بلادرنگ بودن آن است که تمایز آن را با سایر روش‌ها ایجاد می‌کند. درمجموع این روش دارای نوآوری‌های مناسب و نتایج بسیار قابل قبولی می‌باشد.

در [18] روشی هوشمند و مبتنی بر الگوریتم دسته‌بندی‌کننده بیز ساده[6] با تمرکز بر روی الگوریتم ژنتیک برای شناسایی حملات محروم‌سازی از سرویس ارائه شده که شباهت بسیار زیادی از لحاظ نوآوری و نحوه پیاده سازی به روش ارائه شده در این مقاله دارد و فقط از لحاظ الگوریتم یادگیری ماشین به کار رفته در روش ارائه شده، متفاوت است.

۲- روش ارائه شده

همان‌طور که در بخش‌های قبل نیز به آن اشاره شد، روش ارائه‌شده در این تحقیق مبتنی بر یادگیری ماشین و بهینه‌سازی الگوریتم‌های ژنتیک است. در این تحقیق از طبقه‌بند جنگل تصادفی استفاده می‌شود. هدف این است که طبقه‌بند فوق توسط الگوریتم‌های ژنتیک تا حد ممکن بهینه شود و در انتها، دقت شناسایی حملات DDOS توسط این الگوریتم‌ها با یکدیگر مقایسه شده و بهترین الگوریتم انتخاب شود.

۲-۱- روند کلی الگوریتم‌های ژنتیکی

در شکل ۱ یک الگوریتم ژنتیکی استاندارد و در شکل ۲ نمودار گردشی الگوریتم‌های ژنتیکی نشان داده شده است. قبل از این‌که یک الگوریتم ژنتیکی بتواند اجرا شود، ابتدا باید کدگذاری یا نمایش مناسبی برای مسئله موردنظر پیدا شود. همچنین یک تابع برازندگی نیز باید ابداع شود تا به هر راه‌حل کدگذاری شده ارزشی را نسبت دهد.

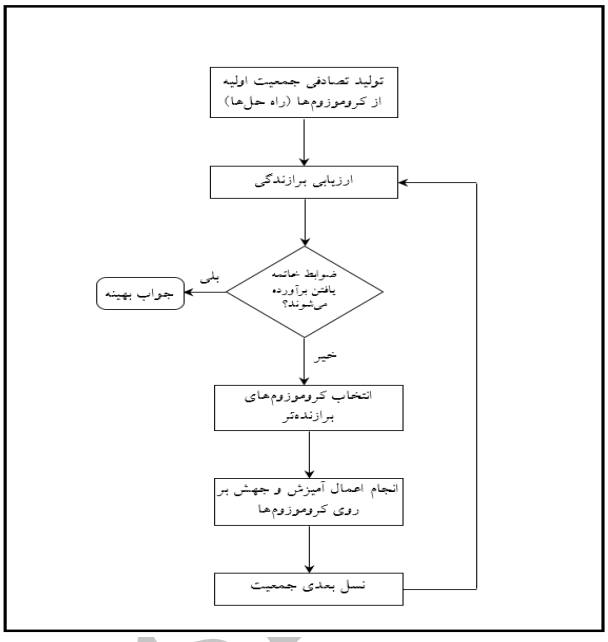
در طی اجرا، والدین برای تولیدمثل انتخاب می‌شوند و با استفاده از عملگرهای آمیزش و جهش باهم ترکیب می‌شوند تا فرزندان جدیدی تولید کنند. این فرآیند چندین بار تکرار می‌شود تا نسل بعدی جمعیت تولید شود. سپس این جمعیت بررسی می‌شود و درصورتی‌که ضوابط همگرایی برآورده شوند، فرآیند فوق خاتمه می‌یابد.

```
BEGIN  /* genetic algorithm */
    Generate initial population
    Compute fitness of each individual

    WHILE NOT finished DO
    BEGIN  /* produce new generation */

        FOR population_size / 2 DO
        BEGIN  /* reproductive cycle */
            Select two individuals from old generation for mating
                /* biased in favor of the fitter ones */
            Recombine the two individuals to give two offspring
            Compute fitness of the two offspring
            Insert offspring in new generation
        END
```

شکل ۱: الگوریتم استاندارد ژنتیک[9]

۲-۱-۱- روش انتخاب تورنمنت

ایده اصلی در روش‌های انتخاب این است که افراد بهتر بر افراد بدتر ترجیح داده شوند، که بهتر و بدتر بودن افراد توسط تابع برازندگی f تعریف می‌شود. روش‌های انتخاب متعددی برای استفاده در الگوریتم‌های ژنتیکی پیشنهاد شده‌اند. یکی از ویژگی‌های خوب روش‌های انتخاب این است که این روش‌ها مستقل از نمایش افراد جمعیت هستند و در آن‌ها تنها مقادیر برازندگی افراد در نظر گرفته می‌شود.



شکل ۲: نمودار گردشی الگوریتم‌های ژنتیک[9]

انتخاب تورنمنت[7] مشابه با انتخاب رتبه‌ای برحسب فشار انتخاب است، امّا ازنظر محاسباتی کاراتر و برای پیاده‌سازی‌های موازی مناسب‌تر است. در این روش، دو فرد از جمعیت به‌صورت تصادفی انتخاب می‌شوند. سپس، یک عدد تصادفی r بین صفر و یک انتخاب می‌شود. k یک پارامتر است واگر r < k باشد، برای مثال اگر ۰/۷۵ باشد، فرد برازنده‌تر و در غیر این صورت فردی که برازندگی کمتری دارد، به‌عنوان والد انتخاب می‌شود. این دو سپس به جمعیت اولیه بازگردانده می‌شوند و دوباره در فرآیند انتخاب شرکت داده می‌شوند.

۲

۲-۱- شرط پایان الگوریتم

در الگوریتم‌های تکاملی غالباً اجرای برنامه برای تعداد نسل‌های از پیش تعیین‌شده‌ای صورت می‌گیرد؛ اما شرط دیگری نیز برای پایان الگوریتم‌های ژنتیک توسط Grefenstette [10] ارائه شده است که آن میزان پراکندگی بیت‌ها درون جمعیت[8] است. این محک نشان‌دهنده میزان همگرا شدن کد اعضای جمعیت است. اگر کد یک عنصر دارای طول یک بیت باشد و به‌صورت $\bar{a}_i(a_{il}, \ldots, a_{it})$ نشان داده شود و μ تعداد اعضای جمعیت است که بین $\{\mu, \ldots, 1\}$ می باشد ، مقدار پراکندگی بیت جمعیت P، b(P) به‌صورت زیر تعریف می‌شود:

$$b(p) = \frac{1}{\mu.1}\sum_{j=1}^{1} Max\{\sum_{i=1}^{\mu}(1-a), \sum_{i=1}^{\mu} a_{ij}\} \quad (۱)$$
$$\in [0.5, 1.0]$$

هراندازه میزان b بزرگ‌تر باشد میزان پراکندگی بیت‌ها درون جمعیت کمتر خواهد بود. در حالت ویژه اگر b(P)=1 باشد به این معنی است که کد همه اعضای جمعیت یکسان است. شرط پایان به‌صورت $b(p) > b_{max}$ تعریف می‌شود که b_{ma} معمولاً ۰/ ≈ ۰/۹۵ است.

۲-۲- الگوریتم جنگل تصادفی

این روش یک فن یادگیری جمعی[9] برای طبقه‌بندی است که مجموعه‌ای از درختان تصمیم گیری[10] را در زمان آموزش می‌سازد و خروجی نهایی، مد و یا میانگین خروجی این درختان می‌باشد.

جنگل تصادفی چالش overfitting در درختان تصمیم معمول را مرتفع می‌سازد. به‌عبارت‌دیگر مجموعه‌داده اولیه به چندین قسمت تقسیم می‌شود و هر قسمت بر روی یکی از درختان آموزش داده می‌شود و خروجی نهایی گرفته می‌شود. اکنون با توجه به سیاست‌های مختلف میانگین یا مد خروجی-های درخت‌ها به‌عنوان خروجی نهایی در نظر گرفته می‌شود[8].

هدفی که در این تحقیق دنبال می‌شود، استفاده از الگوریتم‌های ژنتیک برای بهبود طبقه‌بند است. هدف اصلی در استفاده از الگوریتم‌های ژنتیک، انتخاب ویژگی است. با توجه به این‌که مجموعه‌داده حملات DDoS، عموماً دارای تعداد بسیار زیادی ویژگی هستند، استفاده از روشی مناسب برای انتخاب ویژگی نیز تأثیر بسزایی بر کارایی طبقه‌بند جنگل تصادفی دارد. روال کلی استفاده از الگوریتم‌های ژنتیک برای انتخاب ویژگی از مجموعه ویژگی‌ها به‌صورت زیر است:

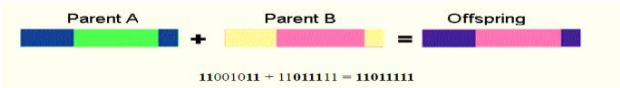
جمعیت اولیه به‌صورت تصادفی تولید می‌شوند. هر نمونه از جمعیت شامل n ژن است که برابر تعداد ویژگی‌ها در مجموعه‌داده است. به‌عبارت‌دیگر هر ژن مشخص می‌کند که آیا ویژگی متناسب با آن در ساخت مدل استفاده شده است یا خیر، اگر استفاده شده باشد مقدار آن ۱ و در غیر این صورت مقدار آن صفر خواهد بود. درنتیجه هر نمونه در جمعیت نشان‌دهنده انتخاب برای ویژگی‌های موجود است. برای هر نمونه در جمعیت جاری، مدل مربوطه ایجاد می‌شود.

بعد از این‌که مدل جنگل تصادفی مربوطه ایجاد شد، این مدل با مجموعه داده ارزیابی[11]، ارزیابی می‌شود و میزان classification error rate آن استخراج می‌شود. جنگل تصادفی ای که مقدار کمتری classification error rate داشته باشد نمونه مناسب‌تر و مطلوب‌تری است.

زمانی که مقدار تابع ارزیابی یا همان classification error rate برای تمام نمونه‌های جمعیت محاسبه شد، الگوریتم ژنتیک نسل بعدی را به‌صورت زیر می‌سازد:

۱) انتخاب نمونه‌ها برای ساخت نسل بعدی با استفاده از Rank selection method که در ادامه توضیح داده می‌شود.
۲) از روش two point crossover برای ساخت فرزندان استفاده می‌شود (شکل ۳)، این روش به‌صورت زیر محاسبه می‌شود:

در این روش در کروموزوم والدین دو نقطه انتخاب می‌شود و هر آنچه میان این دو نقطه قرار دارد جابجا می‌شود تا فرزندان جدید تولید شوند.



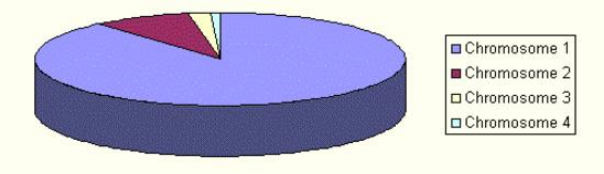
شکل ۳: روش Two-point crossover[11]

۳) در هر فرزند تولیدشده از روش bit level mutation برای جهش استفاده می‌کنیم. در این روش یک بیت به‌صورت تصادفی انتخاب می‌شود و مقدار آن جابجا می‌شود یعنی اگر صفر باشد، یک می‌شود و اگر یک باشد، صفر می‌شود. احتمال انتخاب هر بیت برای جهش در این روش برابر$\frac{1}{l}$ است که l طول کروموزوم است. دو نمونه برتر نگه‌داشته می‌شوند و تمام جمعیت جاری با offspringها جایگزین می‌شوند.

مراحل فوق به‌صورت مکرر و تا زمان رسیدن به حداکثر تعداد تکرارها که ۱۰۰۰ دفعه در نظر گرفته شده، اجرا می‌شوند.

نکته اصلی که در الگوریتم‌های ژنتیک وجود دارد و می‌تواند تأثیر بسزایی در عملکرد آن داشته باشد و مانع از همگرایی سریع[12] شود، روش انتخاب کروموزوم‌های برتر است که در ادامه چندین روش انتخاب معمول که در الگوریتم‌های ژنتیک از آن‌ها استفاده می‌شود، توضیح داده می‌شود.

فرض کنید برای تمام کروموزوم‌ها، چرخ رولت[13] شکل ۴، را به نحوی می‌سازیم، که هر کروموزومی که مقدار تابع fitness بیشتری دارد مساحت بیشتری را به خود اختصاص دهد.



شکل ٤: چرخ رولت کروموزوم‌ها[11]

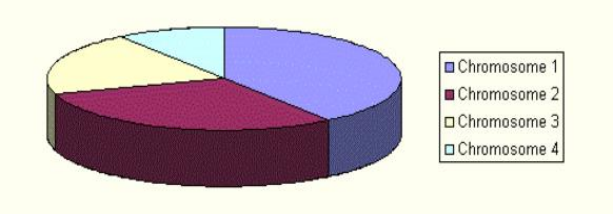
در روش معمول چرخ رولت، کروموزوم با بیشترین مقدار تابع fitness شانس بیشتری برای انتخاب شدن دارد که نحوه انتخاب آن به‌صورت زیر است:

۱) مجموع تمام مقادیر تابع fitness برای هر کروموزوم محاسبه می‌شود و در متغیر S قرار داده می‌شود.

۲) یک مقدار تصادفی r که بین (0,S) قرار دارد انتخاب می‌شود.

۳) بر روی جمعیت کروموزوم‌ها و مقادیر تابع fitness از صفر تا S گذر می‌شود تا زمانی که مقدار جاری از r بزرگ‌تر باشد، در این حالت عمل متوقف می‌شود و کروموزوم مربوطه برگشت داده می‌شود.

بدیهی است که در این حالت، کروموزوم‌ها با مقادیر fitness بیشتر شانس بالاتری برای انتخاب دارند؛ اما اگر یک کروموزوم دارای مقدار fitness بسیار زیادی، مثلاً ۹۰ درصد مجموع کل مقادیر توابع fitness باشد، در این حالت شانس بسیار کمی برای انتخاب دیگر کروموزوم‌ها به وجود می‌آید. در روش Rank Selection برای حل این مشکل، به هر کروموزوم یک مقدار rank اختصاص می‌یابد که ۱ برای بدترین کروموزوم، ۲ برای دومین-بدترین و درنهایت N برای بهترین کروموزوم است. در این حالت، تمام کروموزوم‌ها شانس یکسانی برای انتخاب شدن دارند (شکل ۵)، اما مشکل اصلی این روش در همگرایی کند آن است، زیرا میان بهترین کروموزوم و دیگر کروموزوم‌ها هیچ تفاوتی قائل نشده است.



شکل ٥: چرخ رولت کروموزوم‌ها برای روش Rank Selection

همان‌طور که ملاحظه می‌شود هرکدام از روش‌ها فوق به‌نوعی دارای کمبود و یا نقصی در عملکرد خود هستند. در این پژوهش از روش ارائه شده در [12] استفاده می‌شود که طی آزمایش‌ها و تحلیل‌هایی که بر روی این روش انجام شده است دارای کارایی مناسبی است.

مراحل این الگوریتم به‌صورت زیر است:

۱) جمعیت کروموزوم‌ها را با توجه به مقدار تابع fitness آن‌ها sort می‌کنیم.

۲) جمعیت را به دو قسمت تقسیم می‌کنیم. قسمت با مقادیر fitness بالاتر (HF) و قسمت با مقادیر fitness پایین‌تر (LF) که محل شکستن جمعیت نیز توسط پارامتر m که به‌صورت درصدی بیان می‌شود تعیین می‌گردد.

۳) همیشه والد اول (P1) را از HF و والد دوم (P2) را از LF انتخاب می‌شود.

۴) فرزندان P1 و P2 محاسبه می‌شوند.

۵) فرزندان به انتهای جمعیت اضافه می‌شوند.

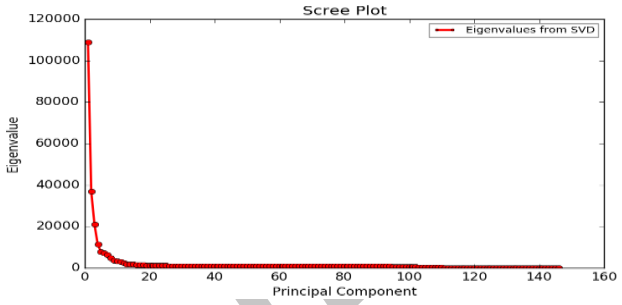
۶) جمعیت مجدد بر اساس مقدار تابع fitness مرتب می‌شود و کروموزوم‌های انتهایی حذف می‌شوند تا طول جمعیت ثابت بماند.

۲-۳- تحلیل روش ارائه شده

از الگوریتم‌های ژنتیک برای شناسایی و انتخاب زیرمجموعه مؤثر از ویژگی‌ها برای طبقه‌بندی استفاده می‌شود. به‌عبارت‌دیگر سعی شده است تا ویژگی‌هایی که بیشترین تأثیر را در جداسازی داده‌ها دارند از میان ویژگی‌های اولیه انتخاب و سپس طبقه‌بندی بر اساس این ویژگی‌ها انجام شود. برای نشان دادن تأثیر ویژگی‌ها از الگوریتمSupport Vector Decomposition [13] استفاده شده است تا برای هر ویژگی در مجموعه‌داده، مقدار ویژه آن محاسبه شود. هرچه مقدار ویژه بیشتر باشد، نشان از تأثیر بیشتر ویژگی مربوطه در جداسازی و یا تعریف داده‌ها دارد. در این تحقیق ازمجموعه‌داده NSL-KDD 2013 [14] استفاده شده است که شامل۴۱ویژگی است وتوسعه‌یافته مجموعه‌داده KDDCup99 [15] می‌باشد.

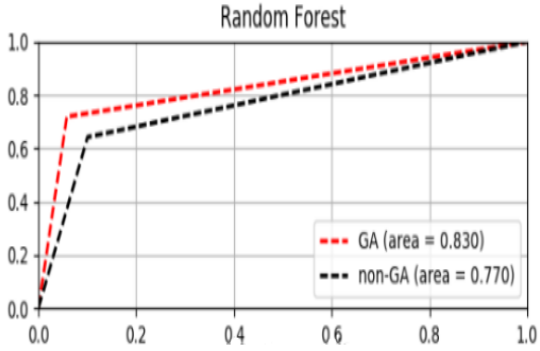
شایان‌ذکر است که در فرآیند پیش‌پردازش، ویژگی‌هایی که دارای مقادیر عددی نیستند، خود به چندین ویژگی عددی شکسته می‌شوند تا بتوان از این ویژگی‌ها نیز در الگوریتم جنگل تصادفی استفاده نمود، درنتیجه پس از پیش‌پردازش تعداد ویژگی‌ها بسیار بیشتر از ۴۱ ویژگی اولیه می‌شود. شکل ۶ مقادیر ویژه برای ویژگی‌های این مجموعه‌داده را به‌صورت نزولی نشان می-

دهد. همان‌طور که در شکل نیز مشخص است، حدود ۲۰ ویژگی از حدود ۱۵۰ ویژگی، پس از انجام پیش‌پردازش‌های لازم در این مجموعه‌داده دارای مقدار ویژه بزرگ‌تر از صفر هستند و این نشان از عدم تأثیر حجم زیادی از ویژگی‌ها، حدود ۱۳۰ ویژگی در تعریف و یا توزیع داده‌ها است، بنابراین در این تحقیق از فن الگوریتم ژنتیک برای شناسایی ویژگی‌های با تأثیر بالا در عملکرد طبقه‌بند استفاده شد.



شکل ٦: مقادیر ویژه برای ویژگی‌های مجموعه‌داده NSL-KDD

در ادامه تأثیر الگوریتم‌های ژنتیک بر نتیجه طبقه‌بندهای در نظر گرفته‌شده بررسی می‌شود. در شکل ۷ برای طبقه‌بند جنگل تصادفی، تفاوت کارایی طبقه‌بند با و بدون اعمال الگوریتم‌های ژنتیک در حالت اولیه مورد بررسی قرار گرفته است. برای ارزیابی عملکرد طبقه‌بند از پارامتر ROC[۱۴][۱۶] استفاده شود که نموداری برحسب نرخ خطای مثبت[۱۵] و نرخ مثبت واقعی[۱۶] است. طبقه‌ای که مساحت سطح زیر نمودار ROC بیشتر باشد نشان‌دهنده این است که به‌عبارتی مربوطه با دقت و احتمال بیشتری جواب صحیح تولید می‌کند و دیگر، هرچه سطح زیر نمودار بیشتر باشد، طبقه‌بند مربوطه کارایی بهتری دارد. همان‌طور که در شکل‌نیز مشخص است، در طبقه‌بند درخت تصمیم در نظر گرفته‌شده، در صورت اعمال الگوریتم‌های ژنتیک شاهد عملکرد بهتری هستیم. با توجه به توضیحات ارائه‌شده، کارایی این روش در بهبود دقت طبقه‌بند درخت تصمیم واضح و روشن می‌شود.



شکل ۷: مقایسه کارایی طبقه‌بند جنگل تصادفی با و بدون اعمال الگوریتم‌های ژنتیک در حالت اولیه

۳- نتایج

۳-۱- مجموعه داده

مجموعه‌داده‌ای که در این تحقیق برای پیاده‌سازی روش ارائه‌شده از آن استفاده می‌شود، مجموعه‌داده NSL-KDD [۱۴] است. این مجموعه‌داده شامل سه فایل اصلی زیر است:

۱) مجموعه‌داده آموزش که شامل ۱۲۵۹۷۳ داده است که هر داده نیز شامل ۴۱ ویژگی می‌باشد.
۲) مجموعه‌داده آزمون اول که شامل ۲۲۵۴۴ داده است.
۳) مجموعه‌داده آزمون دوم که شامل ۱۱۸۵۰ داده است.

باتوجه به این‌که روش ارائه‌شده در این تحقیق برای شناسایی حملات محروم‌سازی از سرویس در بستر رایانش ابری می‌باشد، برای ارزیابی، الگوریتم را به‌طور مستقل و فارغ از نیازمندی‌های محیط رایانش ابری با روش های معمول مقایسه کردیم تا میزان دقت، کارایی و عملکرد روش پیشنهادی در مقایسه با روش‌ها پایه مشخص گردد. به‌عبارت دیگر از لحاظ کردن شرط و یا فرضی که به کلیات مسئله حملات محروم‌سازی از سرویس آسیبی بزند، اجتناب شده است.

۳-۲- پیش‌پردازش مجموعه‌داده

مجموعه‌داده فوق برای آماده شدن برای اجرای الگوریتم درخت تصمیم و الگوریتم‌های ژنتیک، نیاز به فرآیند پیش‌پردازش دارد که در قالب فازهای زیر انجام می‌شود:

۱) حذف مقادیر تهی

برای حذف مقادیر تهی از جایگزین کردن این مقادیر با میانگین مقادیر همان ویژگی در مجموعه‌داده استفاده شده است.

۲) نرمال‌سازی ویژگی‌ها

با توجه به این‌که دامنه مقادیر ویژگی‌ها متفاوت است و اختلاف زیادی با یکدیگر دارد، می‌بایست یکسان‌سازی دامنه این مقادیر برای بهبود عملکرد الگوریتم‌های یادگیری ماشین صورت پذیرد که این امر نیز بر روی مجموعه‌داده اعمال شده است. برای نرمال‌سازی ویژگی‌ها از قالب نرمال‌سازی Z-score [۱۷] استفاده شده است.

۳-۳- پیاده‌سازی روش ارائه‌شده

برای پیاده‌سازی روش ارائه‌شده از زبان برنامه‌نویسی پایتون نسخه ۳.۵ استفاده شده است. مشخصات سخت‌افزاری که الگوریتم بر روی آن اجرا شده است در جدول ۱ آمده است. برای پیاده‌سازی روش از کتابخانه‌های مربوط به الگوریتم‌های یادگیری ماشین، الگوریتم‌های ژنتیک و محاسبات ریاضی ماتریسی که به زبان پایتون هستند، استفاده شده است. لیست این کتابخانه‌ها به همراه توضیحات عملکرد آن‌ها در جدول ۲ آورده شده است.

۳-٤- پارامترهای ارزیابی

برای ارزیابی روش ارائه‌شده از مجموعه پارامترهای ماتریس درهم‌ریختگی[۱۷] استفاده شده است.

جدول۱: مشخصات سخت افزار پیاده سازی

سیستم‌عامل	ویندوز ۷، نسخه ۶۴ بیتی
زبان برنامه‌نویسی	پایتون نسخه ۳.۵.۱
ویرایشگر متن	JetBrains Pycharms 2016
پردازنده	Core i7-4510U
حافظه اصلی	8 GB

جدول۲: کتابخانه های استفاده شده

نام کتابخانه	کاربرد
Scikit-learn	الگوریتم‌های یادگیری ماشین
deap	الگوریتم‌های ژنتیک
numpy	محاسبات ریاضی به‌صورت ماتریسی

۳-۵- ارزیابی روش ارائه شده

در این تحقیق از الگوریتم جنگل تصادفی برای شناسایی حملات محروم سازی از سرویس توزیع شده استفاده شده است. نوآوری اصلی تحقیق در استفاده از الگوریتم‌های ژنتیک برای انتخاب زیرمجموعه بهینه از ویژگی‌های مجموعه‌داده است تا عملکرد الگوریتم جنگل تصادفی بهبود یابد. در جدول های ۳و۴ نتایج ارزیابی الگوریتم فوق با و بدون اعمال الگوریتم‌های ژنتیک اعمال شده بر روی مجموعه داده های KDDTest+ و KDDTest^{-21} آورده شده است.

همان‌طور که در جدول های شماره ۳و۴ نیز مشخص است ، الگوریتم جنگل تصادفی که با الگوریتم‌های ژنتیک بهینه شده‌اند، نسبت به حالت معمول الگوریتم جنگل تصادفی از کارایی و عملکرد بهتری برخوردار هستند. از نظر زمان اجرای الگوریتم در شرایط مختلف نیز مقایسه‌ای صورت گرفته است که در جدول ۵ این نتایج آمده است. البته دقت کنید این زمان، مدت زمانی است که الگوریتم برای آموزش نیاز دارند که زمان آفلاین محسوب می‌شود.

٤- بحث و نتیجه گیری

در این تحقیق، به ارائه روشی مبتنی بر الگوریتم‌های ژنتیک برای شناسایی حملات محروم سازی از سرویس توزیع شده در بستر رایانش ابری پرداخته شد.

رویکرد اصلی در این تحقیق بهینه ساختن عملکرد طبقه‌بند جنگل تصادفی با استفاده از الگوریتم‌های ژنتیک است. به‌عبارت دیگر از الگوریتم-های ژنتیک برای یافتن زیرمجموعه‌ای از ویژگی‌ها استفاده شد که این زیرمجموعه بیشترین دقت در شناسایی حملات محروم سازی از سرویس توزیع شده در الگوریتم درخت تصادفی را سبب می‌شود، درنهایت نیز روش ارائه‌شده بر روی مجموعه‌داده معتبری اجرا و نتایج حاصل‌شده حاکی از عملکرد و دقت بیشتر روش ارائه‌شده دارد.

٤-١- پیشنهادات

با توجه به رشد روزافزون فناوری و حملات سایبری، لزوم استفاده از روش-های مبتنی بر تحلیل داده و آمار اطلاعات می‌تواند نقش مهمی را در کاهش آسیب‌های ناشی از حملات سایبری ایفا کند. ازاین‌رو پیشنهادهای زیر به‌منظور ادامه پژوهش جاری ارائه می‌گردد:

۱) مقیاس‌پذیر بودن

با توجه به بزرگ‌شدن شبکه‌ها و افزایش اطلاعات، نیاز ضروری است تا روش‌های ارائه‌شده قابلیت اعمال و پشتیبانی حجم زیادی از داده‌ها را داشته باشند.

۲) تمرکز بر روی پیش‌پردازش داده

از آنجایی کیفیت داده‌ها، چگونگی ذخیره‌سازی و قالب نمایش آن‌ها در منابع و حوزه‌های مختلف متفاوت است، ادغام و ترکیب آن‌ها برای اجرای الگوریتم‌های یادگیری ماشین دارای چالش‌هایی خواهد بود که می‌تواند زمینه تحقیقات آینده باشد. باتوجه به این‌که داده‌های شبکه معمولا دارای قالب‌ها ناهمگون هستند، این امر می‌تواند کمک شایانی به افزایش دقت شناسایی حملات سایبری نماید.

جدول۳: مقایسه نسخه های مختلف با مجموعه داده آزمون اول

نام روش	KDDTest+			
	Precision	Recall	F-Measure	Accuracy
Random Forest	۰.۷۸	۰.۷۴	۰.۷۴	۰.۷۳۸۶
Random Forest+ GA	۰.۸۳	۰.۷۹	۰.۷۹	۰.۷۸۶۹

جدول٤: مقایسه نسخه های مختلف با مجموعه داده آزمون دوم

نام روش	KDDTest^{-21}			
	Precision	Recall	F-Measure	**Accuracy**
Random Forest	۰.۷۶	۰.۵۴	۰.۵۹	۰.۵۳۵۱
Random Forest + GA	۰.۸۰	۰.۵۴	۰.۵۹	۰.۵۴۴۲

جدول۵: میزان زمان اجرای الگوریتم ها به دقیقه

نام روش	Run Time (min)
Random Forest	۱
Random Forest + GA	۴۵۶

۳) استفاده از فن یادگیری عمیق[۱۸]

یکی از حوزه‌های تحقیقاتی که اخیرا مورد اقبال پژوهشگران قرار گرفته است، مباحث مربوط به یادگیری عمیق است. در مساله تحقیق جاری نیز می-توان از قدرت شبکه‌های عصبی یادگیری عمیق برای استخراج و یادگیری ویژگی استفاده نمود.

سپاسگزاری

از استادان گرامیم آقای دکتر چنگیز دل آرا و خانم دکتر پریسا دانشجو بسیار سپاسگذارم چرا که بدون راهنماییهای ایشان به سرنجام رساندن این پژوهش کاری بسیار مشکل بود.

مراجع

[1] Wang, B., et al., *DDoS attack protection in the era of cloud computing and software-defined networking*. Computer Networks, 2015. 81: p. 308-319.
[2] Tama, B.A. and K.-H. Rhee, *Data Mining Techniques in DoS/DDoS Attack Detection: A Literature Review*. International Information Institute (Tokyo). Information, 2015. 18(8): p. 3739.
[3] Li, X., et al. *DDoS Detection in SDN Switches using Support Vector Machine Classifier*. in 2015 Joint International Mechanical, Electronic and Information Technology Conference (JIMET-15). 2015. Atlantis Press.
[4] Malhi, A.K. and S. Batra, *Genetic-based framework for prevention of masquerade and DDoS attacks in vehicular ad-hocnetworks*. Security and Communication Networks, 2016. 9(15): p. 2612-2626.
[5] Ambusaidi, M.A., et al., *Building an intrusion detection system using a filter-based feature selection algorithm*. IEEE transactions on computers, 2016. 65(10): p. 2986-2998.
[6] Osanaiye, O., et al., *Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing*. EURASIP Journal on Wireless Communications and Networking, 2016. 2016(1): p. 130.
[7] Varma, P.R.K., V.V. Kumari, and S.S. Kumar, *Feature Selection Using Relative Fuzzy Entropy and Ant Colony Optimization Applied to Real-time Intrusion Detection System*. Procedia Computer Science, 2016. 85: p. 503-510.
[8] Liaw, A. and M. Wiener, *Classification and regression by randomForest.* R news, 2002. 2(3): p. 18-22.
[9] Holland, J.H. and J.S. Reitman, *Cognitive systems based on adaptive algorithms.* ACM SIGART Bulletin, 1977(63): p. 49-49.
[10] Grefenstette, J.J., *Optimization of control parameters for genetic algorithms*. IEEE Transactions on systems, man, and cybernetics, 1986. 16(1): p. 122-128.
[11] Mitchell, M., *An introduction to genetic algorithms*. 1998: MIT press.
[12] Ali, E. and E. Elamin, *A proposed genetic algorithm selection method*. 2006.
[13] Pereira, F. and G. Gordon. *The support vector decomposition machine. in Proceedings of the 23rd international conference on Machine learning*. 2006. ACM.
[14] Revathi, S. and A. Malathi, *A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection*. 2013.
[15] Cup, K., Dataset. available at the following website http://kdd.ics.uci.edu/databases/kddcup99/kddcup99. html, 1999. 72.
[16] Hanley, J.A. and B.J. McNeil, *The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology*, 1982. 143(1): p. 29-36.
[17] Cheadle, C., et al., *Analysis of microarray data using Z score transformation. The Journal of molecular*
[18] Mahmodi Derakhsh, A., Daneshjoo, P., and Delara, C., *Using Genetic Algorithm to Improve Bernoulli Naïve Bayes Algorithm in Order to Detect DDos Attacks in Cloud Computing Platform*. International Journal of Sciense and Engineering Investigations, Volume7, Issue72, January 2018.

زیر نویس

1 Denial-of-service attack (DOS)
2 Random Forest
3 Genetic Algorithm
4 Distributed Denial-of-service attack (DDOS)
5 Software Defined Networking (SDN)
6 Bernoulli Naïve Bayes
7 Tournament Selection
8 Bit Diversity
9 Ensemble Learning
10 Decision Trees
11 Validation Dataset
12 Rapid Convergence
13 Roulette wheel
14 Receiver Operating Characteristic
15 False Positive Rate (Fall-out)
16 True Positive Rate (Sensitivity)
17 Confusion Matrix
18 Deep Learning