

## بهینه سازی تشخیص حملات تزریق SQL با استفاده ترکیبی از الگوریتم های جنگل تصادفی و ژنتیک

جواد مرادی<sup>1</sup>، مجید غیوری ثالث<sup>2</sup>

تاریخ پذیرش: 1400/02/12

تاریخ دریافت: 1399/06/11

### چکیده

علی‌رغم تمام تلاش متخصصان امنیتی برای کشف حملات تزریق SQL، اما بر اساس گزارش OWASP، کماکان حمله تزریق SQL به‌عنوان مهم‌ترین و زیان‌بارترین حمله سایبری توسط مهاجمین مورد استفاده قرار می‌گیرد. به‌منظور تشخیص حملات از دو روش مبتنی بر امضاء و مبتنی بر رفتار استفاده می‌شود. روش‌های مبتنی بر امضاء برای حملات شناخته شده کاربرد دارند و روش‌های مبتنی بر رفتار برای تشخیص حملات ناشناخته مناسب هستند. از آنجایی که حملات به روش‌های مختلفی پیاده‌سازی می‌شوند سیستم‌های تشخیص نفوذ مبتنی بر رفتار، کاربرد بیشتری دارند. رفتار را می‌توان با استفاده از روش‌هایی مانند طبقه‌بندی، خوشه‌بندی و غیره تحلیل کرد. یکی از مهم‌ترین الگوریتم‌های طبقه‌بندی، الگوریتم جنگل تصادفی است که دقت بالایی دارد و از طرفی پیاده‌سازی و تفسیر نتایج با استفاده از این الگوریتم به سادگی قابل انجام است. با توجه به بررسی‌های انجام شده دقت الگوریتم جنگل تصادفی به‌شدت وابسته به پارامترهای ورودی آن است. این پارامترها شامل ۹ مورد از جمله تعداد درخت‌ها، عمق آن‌ها، نحوه رأی‌گیری، بهره‌آطلاعاتی و غیره است. تعیین بهینه این پارامترها یک مسئله بهینه‌سازی با فضای حالت بزرگ است. در این پژوهش روشی بر اساس الگوریتم ژنتیک برای تعیین مقادیر بهینه این پارامترها ارائه شده است. در اثر تعیین بهینه پارامترها، نتایج به‌دست‌آمده در مقایسه با حالت پیش‌فرض الگوریتم و سایر تحقیقات، بهبود دقت تشخیص را نشان می‌دهد. نتایج ارزیابی حاکی از آن است که دقت تشخیص نفوذ در روش پیشنهادی، ۹۸٪ بوده است که در مقایسه با الگوریتم جنگل تصادفی با پارامترهای پیش‌فرض حدوداً ۱۱٪ و در مقایسه با پژوهش‌های قبلی ۸٪ دقت تشخیص، افزایش یافته است.

واژه‌های کلیدی: الگوریتم جنگل تصادفی، الگوریتم ژنتیک، حمله تزریق SQL، سیستم تشخیص نفوذ پایگاه داده

<sup>1</sup> کارشناس ارشد رایانش امن، دانشگاه جامع امام حسین (ع)، jmoradi@ihu.ac.ir

<sup>2</sup> استادیار دانشگاه جامع امام حسین (ع)، ghayoori@ihu.ac.ir

## 1. مقدمه

طبقه بندی و پیش بینی را افزایش می دهد، درحالی که بقیه خصوصیات مطلوب یک درخت مثل سادگی تفسیر نتایج را نیز حفظ خواهد شد [17-19].

تحقیقات اخیر نشان داده است که الگوریتم جنگل تصادفی در شناسایی رفتار حملات به خوبی عمل می کند [19] اما در بررسی هایی که ما انجام دادیم مشخص شد تغییر در پارامترهای ورودی الگوریتم جنگل تصادفی تأثیر بسزایی در نتایج خروجی دارد.

به عبارت بهتر، تنظیم پارامترها در الگوریتم جنگل تصادفی مسئله بسیار مهمی است که با تنظیم بهینه آن می توان به نتایج خیلی بهتری نسبت به تحقیقات پیشین دست پیدا کرد؛ اما تنظیم پارامترها با توجه به تعداد و مقادیر زیاد این پارامترها منجر به حالات بسیار زیادی در الگوریتم جنگل تصادفی شده است که مسئله انتخاب بهترین پارامترها را به یک مسئله NP-Complete تبدیل کرده است. در این پژوهش ما سعی داریم با استفاده از الگوریتم فرا ابتکاری ژنتیک، مقدار بهینه پارامترهای الگوریتم جنگل تصادفی را به منظور بالا بردن دقت سیستم تشخیص نفوذ پایگاه داده تعیین کنیم.

## 2. پیشینه تحقیق

هنمتهو و همکارانش [23] در دانشگاه کاکاتیا هند مدلی را بر اساس درخت تصمیم برای جلوگیری از حمله تزریق SQL پیشنهاد دادند. در مدل پیشنهادی، یک پایگاه از انواع حملات تزریق SQL جمع آوری شده و پرس وجوهای جدید بر اساس ساختار درختی تعریف شده با این پایگاه مقایسه شده و اگر تطابقی یافت شد، پرس وجوی موردنظر را به عنوان حمله دسته بندی می کند. در این پژوهش ابتدا یک پیش پردازش بر روی پرس وجوها انجام می شود و در گام بعدی مقدار بهره اطلاعاتی<sup>3</sup> برای نقطه شکست پرس وجوها محاسبه می شود و درخت ها را بر این اساس می سازند. در ادامه هر پرس وجوی جدید، به درخت تصمیم گیری نگاشت می شود و طبقه بندی آن مشخص می شود. از جمله مشکلات این روش می توان به موارد زیر اشاره

تعداد حملات سایبری به صورت چشمگیری در حال افزایش است به طوری که بر اساس گزارش استاتیسنا در سال 2019، 668 میلیون حمله سایبری رخ داده است [1].

بر اساس آخرین گزارش های مرکز تحقیقاتی OWASP، کماکان حملات تزریق جزء شایع ترین و خطرناک ترین حملات بر روی برنامه های تحت وب هستند. حملات تزریق شامل تزریق SQL، XML، ORM، Command و غیره است که یکی از مهم ترین و زیان بارترین آن ها حمله تزریق SQL است. در حمله تزریق SQL مهاجم با تزریق پرس وجوهای پایگاه داده در فیلدهای ورودی برنامه، می تواند داده های حساس از پایگاه داده را بخواند، تغییر دهد، عملیات مدیریتی مانند خاموش کردن پایگاه داده انجام دهد و در برخی موارد فرمان هایی را روی سیستم عامل اجرا کند [2].

یکی از راه کارهای تشخیص و جلوگیری از این حمله، استفاده از سیستم تشخیص نفوذ پایگاه داده است. سیستم های تشخیص نفوذ در دو دسته کلی، مبتنی بر امضاء و مبتنی بر رفتار تقسیم می شوند. در رویکرد مبتنی بر امضاء، الگوهای نفوذ توسط افراد خیره و متخصص تشخیص داده می شود و حملات در چهارچوب این الگوها شناسایی می شوند. ضعف این رویکرد در عدم تشخیص حملات جدید است و همچنین نیازمند به روزرسانی مداوم پایگاه داده امضاءها می باشد؛ اما در رویکرد دوم، سیستم تشخیص نفوذ با ایجاد الگوهای رفتار عادی کاربران، حملات را شناسایی می کند. این روش برای پیدا کردن حملات ناشناخته در پایگاه داده مناسب است. رفتارهای کاربران را می توان با استفاده از روش هایی مانند درخت تصمیم، جنگل تصادفی، نزدیک ترین همسایه، شبکه های عصبی و غیره شناسایی کرد. الگوریتم جنگل تصادفی در واقع یک الگوریتم ترکیبی<sup>2</sup> می باشد. در این روش مجموعه یا جنگلی از درختان مورد استفاده قرار می گیرد. استفاده از مجموعه ای از درخت ها صحت

<sup>3</sup> Information\_Gain

<sup>2</sup> Ensemble



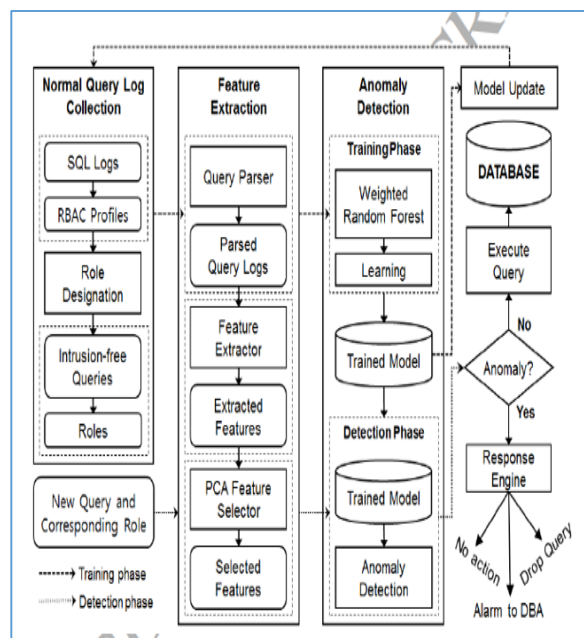
الگوریتم جنگل تصادفی (درختان وزن دار) به منظور بهینه سازی این الگوریتم استفاده شده است، در صورتی که می توان از سایر پارامترهای الگوریتم جنگل تصادفی مانند تعداد درختان، عمق درختان، بهره اطلاعاتی و غیره به منظور بهینه سازی کامل این الگوریتم استفاده کرد. ایراد دوم مربوط به استفاده نادرست از الگوریتم PCA است. در این مقاله برای استفاده از الگوریتم PCA به منظور کاهش ابعاد، باید پرس و جوها به مقادیر عددی تبدیل شوند که این کار نهایتاً منجر به کدگذاری پرس و جو به مقادیر عددی می شود و باعث مدل سازی نادرست توسط الگوریتم جنگل تصادفی می شود.

### 3. روش پیشنهادی

با توجه به موارد ذکر شده در پیشینه تحقیق، تصمیم گرفتیم در گام اول به منظور پیش پردازش مناسب از ایده کار دبا براتا و همکارانش [25] برای نرمال کردن پرس و جوها استفاده کرده و در گام دوم به منظور حداکثر کارایی الگوریتم جنگل تصادفی در مسئله تشخیص حملات تزریق SQL تمامی پارامترهای الگوریتم جنگل تصادفی را به روش سیستماتیک و نه به روش آزمون و خطا، با استفاده از الگوریتم ژنتیک، بهینه سازی کنیم.

در روش پیشنهادی، یک سامانه تشخیص نفوذ مبتنی بر رفتار ارائه کرده ایم. در این مدل، از الگوریتم جنگل تصادفی به عنوان موتور تحلیل گر سیستم تشخیص نفوذ و الگوریتم فرا ابتکاری ژنتیک به منظور بهینه سازی الگوریتم جنگل تصادفی استفاده کرده ایم. دلیل اصلی انتخاب الگوریتم جنگل تصادفی برای موتور تحلیلگر سیستم تشخیص نفوذ، کارایی بالا در عین سادگی این روش می باشد. در الگوریتم جنگل تصادفی 9 پارامتر قابل تنظیم دارد که این پارامترها عبارتند از: تعداد درخت<sup>6</sup>، بهره اطلاعاتی<sup>7</sup>، حداکثر عمق درختها<sup>8</sup>، حداقل سود<sup>9</sup>، حداقل سائز برگها<sup>10</sup>، انجام پردازش<sup>11</sup>، حدس زدن حد زیرمجموعه<sup>12</sup>، نحوه

الگوریتم جنگل تصادفی با رأی گیری وزنی<sup>5</sup> و در کنار آن استفاده از الگوریتم PCA به عنوان یک تکنیک انتخاب ویژگی را پیشنهاد نمودند.



شکل 1: معماری سیستم تشخیص نفوذ پایگاه داده مبتنی الگوریتم جنگل تصادفی [19]

معماری روش مذکور در شکل 1 نمایش داده شده است. در این پژوهش برای هر نقش در پایگاه داده یک پروفایل به عنوان رفتار عادی ایجاد کرده و تراکش هایی که با پروفایل های ساخته شده مطابقت نداشته باشد به عنوان حمله طبقه بندی می شود. برای این کار در مرحله اول، ورودی های کاربران عادی از پایگاه داده جمع آوری شده و پروفایل های متناسب هر نقش ساخته می شود. در مرحله دوم از طریق تجزیه کننده (پارسر)، ویژگی های پرس و جوها را استخراج کرده و این ویژگی ها را معادل سازی عددی می کردند، سپس با استفاده از الگوریتم PCA ابعاد ویژگی ها را کاهش داده و در مرحله آخر با استفاده از الگوریتم جنگل تصادفی مدل سازی را به منظور تشخیص آنومالی ساختند. مدل پیشنهادی فوق دو ایراد مهم دارد، اول اینکه فقط از یک پارامتر

<sup>9</sup> Minimal Gain

<sup>1</sup> Minimal Leaf Size 0

<sup>1</sup> Apply Pruning 1

<sup>1</sup> Guess Subset Ratio 2

<sup>5</sup>WRF

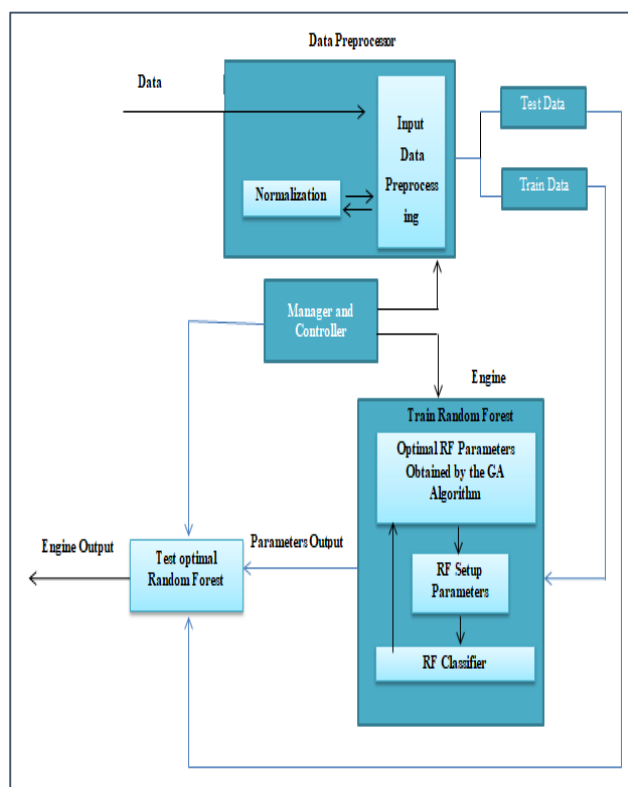
<sup>6</sup> Number of Trees

<sup>7</sup> Criterion

<sup>8</sup> Maximal Depth

1. مؤلفه پیش پردازنده<sup>14</sup>
2. مؤلفه موتور تحلیلگر مبتنی بر الگوریتم جنگل تصادفی<sup>15</sup>
3. مؤلفه مدیر و کنترل کننده<sup>16</sup>

در شکل 2 شمای کلی روش پیشنهادی ترسیم شده است.



شکل 2: معماری طرح پیشنهادی

در این سیستم، ورودی شامل مجموعه‌ای از پرس وجوهای پایگاه داده می‌باشد که به صورت رکوردی در جریان هستند. این رکوردهای وارد مؤلفه‌ی اول، یعنی مرحله پیش پردازش می‌شوند. در مرحله پیش پردازش، پرس وجوها نرمال شده که خروجی حاصل، شامل پرس وجوهایی با شکل یکسان می‌باشد. سپس پرس وجوهای نرمال شده را به مؤلفه تحلیلگر مبتنی بر الگوریتم جنگل تصادفی می‌فرستند. تحلیلگر بر اساس فرمان دریافتی از سوی مدیر و کنترل کننده مرکزی، به ترتیب ابتدا پارامترهای الگوریتم جنگل تصادفی را با استفاده از الگوریتم ژنتیک تعیین می‌کند. سپس با دریافت هر پرس وجو از مجموعه

ی رأی گیری درختان<sup>13</sup> و اجرای موازی الگوریتم. بنابراین تعداد حالات پارامترهای قابل تنظیم 8,000,000 حالت است که این عدد (تعداد حالات قابل تنظیم) از ضرب تعداد حالات تک تک پارامترها به دست آمده است. تعداد حالات هر کدام از پارامترها عبارتند از:

Number of Trees = 100  
 Criterion = 5  
 Maximal Depth = 10  
 Minimal Gain = 10  
 Minimal Leaf Size = 10  
 Apply Pruning = 2  
 Guess Subset Ratio = 2  
 Voting Strategy = 2  
 parallel execution = 2

$$100 \times 5 \times 10 \times 10 \times 10 \times 2 \times 2 \times 2 \times 2 = 8000000$$

با بررسی اکثر مدل‌هایی که از الگوریتم جنگل تصادفی استفاده کرده‌اند، متوجه شدیم که این پارامترها به صورت پیش فرض و با به صورت سعی و خطا توسط مدیر تنظیم شده‌اند. در مدل پیشنهادی قصد داریم که این پارامترها را به صورت سیستماتیک تنظیم کنیم به نحوی که سیستم تشخیص نفوذ حداکثر کارایی را داشته باشد.

به منظور بهینه‌سازی پارامترهای ارائه شده، استفاده از الگوریتم فرا ابتکاری ژنتیک را پیشنهاد کردیم. الگوریتم ژنتیک به عنوان یکی از روش‌های مؤثر جهت حل مسائل بهینه‌سازی یا مسائلی با فضای جستجوی بزرگ شناخته شده است، در واقع فرا ابتکاری ژنتیک برای تعیین پارامترهای الگوریتم جنگل تصادفی، نسبت به سایر الگوریتم‌های فرا ابتکاری دیگر، اولاً برای حل مسائل گسسته و غیرخطی با فضای جستجوی بزرگ بسیار کارا می‌باشد، ثانیاً به علت خصلت تصادفی آن، مشکل گیرکردن در بهینه‌های محلی را به شدت کاهش می‌دهد، ثالثاً پیاده‌سازی آن ساده بوده و نیازی به روال‌های پیچیده حل مسئله ندارد. روش پیشنهاد شده متشکل از 3 مؤلفه اصلی است. این مؤلفه‌ها عبارتند از:

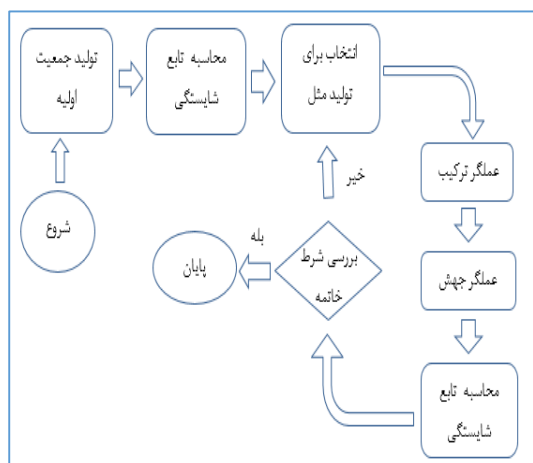
<sup>1</sup> Random Forest Based Analyzer

<sup>1</sup> Manager and Controller <sup>6</sup>

<sup>1</sup> Voting Strategy <sup>3</sup>

<sup>1</sup> Data Preprocessor <sup>4</sup>

- داده‌ی موردنظر، بر اساس تجربیات کسب شده در فاز آموزش، نرمال یا حمله بودن ورودی را مشخص می‌نماید.
- ۱,۴ مؤلفه پیش پردازنده
- وظیفه‌ی پیش پردازش دریافت پرس وجوهای مربوط به پایگاه داده و انجام عمل نرمال سازی بر روی آن‌ها می‌باشد. از آنجایی که بردارهای حمله تزریق SQL، به صورت مختلفی پیاده سازی می‌شود، اگر بتوان بردارهای حمله به یک شکل یکسان تبدیل نمود، انجام تحلیل و تشخیص حملات به راحتی انجام می‌شود. یکی از تحقیقات خوب در زمینه پیش پردازش تراکنش های پایگاه داده، ایده کار دبابارتا و همکارانش [۲۵] می‌باشد که در طی آن تمامی پرس وجوی پایگاه داده را به شکل یکسان تبدیل می‌کند. در جدول ۱ به نحوه نرمال کردن کاراکترهای خاص و در جدول ۲ به نحوه نرمال کردن کلمات و رشته های مورد استفاده در عبارت های SQL اشاره شده است. هدف از این کار تأمین ورودی مناسب برای مرحله بعدی، یعنی یادگیری مدل می‌باشد.
- ۲,۴ تحلیلگر مبتنی بر الگوریتم جنگل تصادفی
- در این مرحله مهم ترین مؤلفه این معماری قرار دارد که بخش موتور تحلیلگر<sup>۱۷</sup> آن است. این مرحله وظیفه پردازش مجموعه داده و در نهایت دسته بندی آن‌ها را بر عهده دارد.
- در الگوریتم جنگل تصادفی تعیین پارامترهای بهینه در عملکرد این الگوریتم بسیار حائز اهمیت است؛ برای رسیدن به این هدف در این مقاله، از الگوریتم بهینه سازی ژنتیک برای تعیین پارامترهای الگوریتم جنگل تصادفی استفاده کرده ایم. این الگوریتم با در نظر داشتن تمام ابعاد مسئله به مرور زمان، فضای جستجوی خود را به سمت بهترین راه حل‌ها محدود می‌کند که این ویژگی منحصر به فرد باعث کارایی بالای این الگوریتم در عین حفظ کارایی و دقت می‌شود. همان طور که در شکل 3 مشاهده می‌شود چرخه الگوریتم ژنتیک به شرح زیر است.
1. تولید جمعیت اولیه
  2. محاسبه تابع شایستگی
3. انتخاب دو کروموزوم از جمعیت به عنوان والد
  4. اجرای عملگر ترکیب به منظور تولید فرزند.
  5. در صورت مهیا بودن شرایط، اجرای عملگر جهش
  6. محاسبه تابع شایستگی فرزندان
  7. بررسی شرط خاتمه
    - اگر شرط خاتمه صحیح بود اجرای الگوریتم متوقف می‌شود و بهترین راه حل انتخاب می‌شود.
    - اگر شرط خاتمه صحیح نبود به گام 2 برگرد.



شکل 3: فلوچارت الگوریتم ژنتیک

در حل هر مسئله ای توسط الگوریتم ژنتیک باید الزامات و توابع این الگوریتم من جمله شیوه کدگذاری، تابع شایستگی، تابع انتخاب، عملگر ترکیب، عملگر جهش و شرط خاتمه الگوریتم تبیین شود. در ادامه به تشریح هر بند پرداخته شده است.

#### ۱,۲,۴ شیوه کدگذاری

الگوریتم ژنتیک بجای اینکه بر روی پارامترها یا متغیرهای مسئله کار کند، با شکل کد شده آن‌ها سروکار دارد. روش های کدگذاری متداول متغیرها در الگوریتم ژنتیک عبارت اند از

<sup>1</sup> Analyzer

کدگذاری باینری<sup>18</sup>، کدگذاری جهشی<sup>19</sup> کدگذاری ارزشی<sup>20</sup> و کدگذاری درختی<sup>21</sup>.

در روش پیشنهادی از کدگذاری جهشی استفاده می‌کنیم و هر پارامتر را معادل یک ژن در نظر می‌گیریم که ۹ ژن در کنار یکدیگر، یک کروموزوم را تولید می‌کند.

### ۲.۲.۴ تابع شایستگی

زمانی که در فضای مسئله به دنبال بهترین جواب‌ها هستیم باید معیاری را به منظور مناسب بودن جواب تعریف کنیم. در الگوریتم ژنتیک، معیاری که این کار را انجام می‌دهد تابع شایستگی<sup>22</sup> م دارد. ورودی تابع شایستگی، کروموزوم‌ها و خروجی تابع، میزان دقت مدل می‌باشد. می‌توان از محاسبه‌ی دقت مدل به عنوان، تابع شایستگی استفاده کرد که هدف غایی الگوریتم ژنتیک، بیشینه کردن تابع شایستگی می‌باشد. دقت مدل، حاصل تعداد رکوردهایی که به درستی توسط مدل پیشنهادی تشخیص داده شده (TP + TN) تقسیم بر تعداد کل رکوردها (TP + TN + FP + FN) می‌باشد. دقت مدل را با استفاده از فرمول ۱ محاسبه می‌شود.

$$\text{Accuracy} = \frac{TP+TN}{(TP+FN)+(TN+FP)} \quad \text{فرمول ۱}$$

در ابتدا از فضای حل مسئله، ۱۰۰ کروموزوم را به صورت تصادفی انتخاب می‌کنیم که این کروموزوم‌های انتخابی، جمعیت اولیه را تشکیل می‌دهند. سپس تابع شایستگی را بر روی کروموزوم‌های جمعیت اولیه اجرا می‌کنیم و میزان شایستگی هر کروموزوم را محاسبه کرده و در گام بعدی بر اساس تابع انتخاب، دو کروموزوم را به عنوان والد برای عمل ترکیب و جهش انتخاب می‌کنیم.

### ۳.۲.۴ تابع انتخاب

همان‌طور که گفته شد پس از محاسبه میزان شایستگی کروموزوم‌ها باید دو کروموزوم را به عنوان والد برای

عملیات ترکیب و جهش انتخاب کنیم. در اینجا از مکانیزم چرخ رولت به عنوان تابع انتخاب استفاده کرده‌ایم. مهم‌ترین مزیت استفاده از چرخ رولت این است که هر کروموزومی از جمعیت، شانس انتخاب شدن به عنوان والد را دارد، در واقع در مکانیزم چرخ رولت هر یک از کروموزوم‌ها بسته به میزان مناسب بودنش (بر اساس تابع شایستگی) احتمال انتخاب شدنش وجود دارد. به عبارت دیگر هر چه یک کروموزوم مقدار تابع شایستگی‌اش بیشتر باشد احتمال انتخاب شدنش برای تولید نسل بعدی بیشتر است و برعکس هر چه مقدار تابع شایستگی کروموزوم کم‌تر باشد، احتمال انتخاب شدنش برای تولید نسل بعدی کمتر است. شیوه پیاده‌سازی چرخ رولت به این صورت است که با توجه به مقادیر تابع شایستگی هر کروموزوم، احتمال انتخاب کروموزوم بر اساس فرمول ۲ محاسبه می‌شود.

$$\text{Probability} = \frac{\text{Fitness}(\text{chromosome } X)}{\text{Sum Fitness (All chromosome)}} \quad \text{فرمول ۲}$$

با توجه به اینکه مجموعه احتمالات کروموزوم‌ها برابر ۱ است در قدم بعدی احتمال انتخاب کروموزوم‌ها را بر روی یک بردار به اندازه ۱ نگاهت می‌دهیم و در گام آخر یک عدد تصادفی بین ۰ تا ۱ تولید می‌کنیم. این عدد در هر بازه‌ای قرار بگیرد یعنی آن کروموزوم انتخاب شده است.

در گام دوم برای جلوگیری از حذف بهترین کروموزوم‌های نسل قبل از خاصیت نخه‌گرایی استفاده کرده‌ایم و همیشه بهترین جواب نسل قبل را بدون هیچ تغییری به نسل جدید منتقل می‌کنیم.

### ۴.۲.۴ عملگر ترکیب

<sup>2</sup> Tree Encoding 1

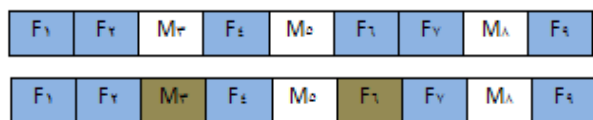
<sup>22</sup>fitness function

<sup>1</sup> Binary Encoding 8

<sup>1</sup> Permutation Encoding 9

<sup>2</sup> Valu Encoding 0

باشد ژن تغییر نمی کند. در شکل 8-4 نمونه ای از یک عمل جهش را مشاهده می کنید.



شکل 4: نمونه ای از عملگر جهش

#### ۶,۲,۴ شرط خاتمه الگوریتم

برای اینکه تشخیص دهیم چه موقع الگوریتم از اجرا متوقف شود، از شیوه های مختلفی می توان استفاده کرد. به عنوان نمونه تولید نسل ها بیشتر  $N$  (مثلاً ۱۰۰۰ نسل) شود یا اگر در طی  $Y$  نسل (مثلاً ۵۰ نسل) حداکثر تابع برازندگی تغییر نکرد را به عنوان شرط خاتمه در نظر گرفت. در این پژوهش به علت بار زمان پردازش سریع تر و همچنین بالا رفتن ضریب دقت مدل، شرط خاتمه الگوریتم را این گذاشتیم که اگر در طی ۵۰ نسل حداکثر تابع برازندگی تغییر نکند.

#### ساخت مدل مبتنی بر الگوریتم جنگل تصادفی

بعد از تعیین پارامترهای الگوریتم جنگل تصادفی، وارد روال آموزش الگوریتم جنگل تصادفی می شویم. در این روال، الگوریتم جنگل تصادفی که شامل مجموعه ای از درختان تصمیم است بر اساس پارامترهای تعیین شده توسط الگوریتم ژنتیک، به یک جنگل بهینه برای آموزش می رسیم که این کار در نهایت منجر به افزایش نرخ تشخیص صحیح در سیستم تشخیص نفوذ می شود.

#### ۳,۴ مدیر و کنترل کننده

در مؤلفه سوم از معماری سیستم تشخیص نفوذ پایگاه داده، مدیر و کنترل کننده قرار دارد. در این مؤلفه مقاردهی اولیه یک سری پارامترها، مدیریت فضای کاربر و اجرای فرمان ها صادر شده از سوی کاربر با برقراری ارتباط با مؤلفه های سیستم می باشد. لذا مشخص نمودن مقدار تابع شایستگی و شرط خاتمه الگوریتم

این عملگر اطلاعات دو کروموزوم والد را گرفته و با ترکیب آن ها یک جفت کروموزوم جدید تولید می کند. در واقع در جریان عمل ترکیب، بخش هایی از کروموزوم ها با یکدیگر تعویض می شوند. این موضوع باعث می شود که فرزندان، ترکیبی از خصوصیات والدین خود را به همراه داشته باشند و از طرفی دقیقاً مشابه یکی از والدین نباشند. به عبارت بهتر هدف عملگر ترکیب تولید فرزند جدید می باشد به این امید که خصوصیات خوب دو موجود در فرزندشان جمع شده و موجود بهتری را تولید کنند. عملگر ترکیب می تواند به صورت تک نقطه ای، چندنقطه ای و یکنواخت باشد.

در میان روش های فوق روش تک نقطه ای دارای تنوع ژنتیکی کمتری نسبت به سایر روش ها می باشد و از طرفی ترکیب یکنواخت دارای تنوع ژنتیکی بیشتری نسبت به ترکیب های تک نقطه ای و چندنقطه ای می باشد به همین دلیل این نوع ترکیب در جمعیت هایی که اعضای کمی دارند اثر بهتری دارد تا جمعیت هایی که تعداد اعضای زیادی دارند؛ بنابراین با توجه به فضای مسئله، ما عملگر چندنقطه ای را انتخاب کردیم.

#### ۵,۲,۴ عملگر جهش

مهم ترین وظیفه عملگر جهش جلوگیری از قرار گرفتن الگوریتم ژنتیک در بهینه های محلی می باشد. جهش نباید زیاد صورت بگیرد زیرا در این صورت الگوریتم ژنتیک به جستجوی کاملاً تصادفی تبدیل خواهد شد. در الگوریتم ژنتیک بعد از اینکه یک عضو در جمعیت جدید به وجود آمد، هر ژن آن با احتمال جهش<sup>23</sup> جهش می یابد. احتمال جهش  $P_{Mutation}$  مقداری است که توسط کاربر تعیین می شود. در الگوریتم استاندارد ژنتیک مقدار این پارامتر، بسیار کوچک، مثل  $P_{Mutation} = 0.01$  یا حتی  $P_{Mutation} = 0.001$  در نظر گرفته می شود. در فرزندی که توسط عملگر ترکیب به وجود آمده است، به ترتیب مقداری تصادفی بین ۰ و ۱ به هر ژن اختصاص می یابد. اگر این مقدار اختصاص داده شده از  $P_{Mutation}$  کمتر باشد، ژن جهش می یابد و اگر بیشتر

<sup>23</sup>Presumption Mutation -  $P_m$

ژنتیک، برای تعیین پارامترها توسط کاربر انجام می‌گیرد که لازم است همان ابتدا تعیین شوند.

#### 4. نتایج

##### مجموعه داده

با توجه به این که مجموعه داده مناسبی برای ارزیابی طرح پیشنهادی پیدا نکردیم، تصمیم بر این شد که مجموعه داده را خودمان تولید کنیم. برای این منظور ابتدا به بررسی ابزارهای معتبر آسیب‌پذیری تزریق SQL پرداختیم و به صورت ویژه بر روی ابزار SQL Map که کد باز<sup>24</sup> می‌باشد تمرکز کردیم و با مراجعه به کد و تنظیمات نرم‌افزار توانستیم بردارهای حمله تزریق SQL را از این نرم‌افزار استخراج کنیم با این کار توانستیم تعداد ۲۰۰ بردار حمله به دست بیاوریم. در ادامه به چند مورد از این بردارهای حمله اشاره شده است.

- 1- [RANDNUM] = [RANDNUM]
- 2- NOT (RANDNUM)=[RANDNUM]
- 3- ELT((RANDNUM)=[RANDNUM], [ORIGVALUE])
- 4- RLIKE SLEEP([SLEEPTIME])
- 5- SLEEP([SLEEPTIME])

##### تولید پرس‌وجوهای نرمال

به منظور تولید پرس‌وجوهای نرمال، یک خزش‌گر پیاده‌سازی کردیم و برای خوراک اولیه خزش‌گر با دادن کلمات کلیدی مانند SQL, QUERY, TSQL و غیره صفحاتی که حاوی این کلمات کلیدی بوده را پیدا کرده و از میان محتوای صفحات دریافت شده با استفاده از چند عبارت منظم به استخراج پرس‌وجوهای موجود در آن‌ها پرداختیم. با این کار توانستیم تعداد 4 هزار پرس‌وجوی نرمال غیر تکراری تهیه کنیم.

##### تولید پرس‌وجوهای حمله

از بین 4 هزار پرس‌وجوی مرحله قبل یک لیست ۱۲۰ تایی از پرس‌وجوهای نرمال دارای قسمت Where Clause تهیه کردیم که تکراری هم نیستند و در گام بعدی با ترکیب بردارهای حمله

با پرس‌وجوهای نرمال توانستیم در مجموع، 2 هزار پرس و جوی حمله غیر تکراری تولید کنیم.

##### پیش‌پردازش و نرمال‌سازی مجموعه داده‌ها

قبل از تحلیل مجموعه داده توسط الگوریتم جنگل تصادفی بهینه‌شده، مجموعه داده را بر اساس روش دبابارتا و همکارانش [۲۵] نرمال کردیم. در واقع با این کار تمامی پرس‌وجوها به شکل یکسان تبدیل شدند که این کار منجر به برقراری نظمی خاص در مجموعه داده شده که سرعت و دقت الگوریتم جنگل تصادفی را بالا برد.

##### آزمایش و ارزیابی

در پیاده‌سازی روش پیشنهادی، قبل از اجرای الگوریتم ژنتیک بر روی الگوریتم جنگل تصادفی، برخی از پارامترهای الگوریتم ژنتیک مانند شرط خاتمه و تابع انتخاب که از بقیه پارامترهای الگوریتم ژنتیک مهم‌تر هستند را تعیین کرده و مابقی پارامترهای این الگوریتم را در حالت پیش‌فرض اجرا کرده‌ایم. نتایجی که در ادامه نشان داده شده، در ۳ حالت شرط خاتمه را بررسی کردیم. حالت اول حداکثر تعداد نسل ۱۰۰۰ باشد. حالت دوم تابع شایستگی بعد از ۲۰ نسل بهبودی نداشته باشد و حالت سوم بعد از ۵۰ نسل تابع شایستگی بهبودی نداشته باشد.

جدول 2: تعیین شرط خاتمه در الگوریتم ژنتیک

معیار	دقت	زمان
شرط خاتمه حلقه تولید حداکثر 1000 نسل	93/56	7m:15s
عدم بهبود تابع شایستگی پس از 20 نسل	90/83	2m:16s
عدم بهبود تابع شایستگی پس از 50 نسل	92/74	3m:28s

با توجه به دقت و مدت زمان طی شده تصمیم گرفتیم شرط خاتمه الگوریتم ژنتیک را بر روی عدم بهبودی تابع شایستگی در ۵۰ نسل بگذاریم.

به منظور تنظیم تابع انتخاب، روش چرخ رولت و حالت پیش‌فرض را امتحان کردیم.

جدول 3: تعیین تابع انتخاب در الگوریتم ژنتیک

معیار	دقت	زمان
حالت پیش فرض	98/60	14m:48s
روش چرخ رولت	98/60	12m:56s

با توجه به اینکه دقت یکسان است ولی مدت زمان چرخ رولت کم تر است، در نتیجه از تابع انتخاب چرخ رولت استفاده کردیم. در موارد فوق شرط خاتمه و تابع انتخاب الگوریتم ژنتیک را تنظیم کردیم، در ادامه الگوریتم ژنتیک را بر روی الگوریتم جنگل تصادفی به منظور تعیین پارامترهای بهینه اجرا کردیم و پارامترهای زیر به عنوان پارامترهای بهینه تشخیص داده شده است.

```
Random Forest.number_of_trees = 21
Random Forest.criterion = accuracy
Random Forest.maximal_depth = 19
Random Forest.minimal_gain = 40.0
Random Forest.minimal_leaf_size = 22
Random Forest.apply_pruning = false
Random Forest.guess_subset_ratio = true
Random Forest.voting_strategy = confidence vote
Random Forest.enable_parallel_execution = true
```

شکل 5: مقدار پارامترهای بهینه

در ادامه، روش پیشنهادی را با مقاله [۱۹] و اجرای الگوریتم در حالت پیش فرض را مورد بررسی و مقایسه قرار دادیم.

جدول 4: مقایسه روش پیشنهادی با سایر روش ها

معیار	accuracy	precision	Recall	f_measure	Time
الگوریتم اجرای پارامترهای پیش فرض	87/52	99/66	62/75	76/99	8m:12s
پیاپی سازی مقاله مرجع [29]	90/30	97/59	72/70	83/25	1h:22m:30s
روش پیشنهادی	98/32	97/75	99/77	98/75	3h:10m:18s

## 5. مراجع (References)

- [۱] <https://www.statista.com>
- [۲] <https://www.owasp.org>
- [۳] Umar, Kabir, Abu Bakar Sultan, Hazura Zulzalil, Novia Admodisastro, and Mohd Taufik Abdullah. "Formulation of SQL Injection Vulnerability Detection as Grammar Reachability Problem." In ۲۰۱۸ International Conference on Information and

- [13] N. Shone, T. Nguyen Ngoc, V. Dinh Phai, Q. Shi, "A Deep Learning Approach to Network Intrusion Detection", *TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE*, VOL. ۲, NO. ۱, FEBRUARY, IEEE ۲۰۱۸.
- [14] Tang, TA, Mhamdi, L, McLernon, D et al, "Deep Learning Approach for Network Intrusion Detection in Software Defined Networking", *International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Fez, Morocco. IEEE, ۲۰۱۶.
- [15] Bockermann, Christian, Martin Apel, and Michael Meier. "Learning sql for database intrusion detection using context-sensitive modelling." In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. ۱۹۶-205. Springer, Berlin, Heidelberg, ۲۰۰۹.
- [16] Resende, Paulo Angelo Alves, and André Costa Drummond. "A survey of random forest based methods for intrusion detection systems." *ACM Computing Surveys (CSUR)* ۵۱, no. ۳ (۲۰۱۸): ۴۸.
- [17] Vorobeva, Alisa A. "Influence of features discretization on accuracy of random forest classifier for web user identification." In *۲۰۱۷ ۲۰th Conference of Open Innovations Association (FRUCT)*, pp. ۴۹۸-504. IEEE, ۲۰۱۷.
- [18] Farnaaz, Nabila, and M. A. Jabbar. "Random forest modeling for network intrusion detection system." *Procedia Computer Science* ۸۹ (۲۰۱۶): ۲۱۳-217.
- [19] Ronao, Charissa Ann, and Sung-Bae Cho. "Mining SQL queries to detect anomalous database access using random forest and PCA." In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. ۱۵۱-160. Springer, Cham, ۲۰۱۵.
- [20] Ronao, Charissa Ann, and Sung-Bae Cho. "Anomalous query access detection in RBAC-administered databases with random forest and PCA." *Information Sciences* ۳۶۹ (۲۰۱۶): ۲۳۸-250.
- [21] Aung, Yi Yi, and Myat Myat Min. "An analysis of random forest algorithm based network intrusion detection system." In *۲۰۱۷ ۱۸th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. ۱۲۷-132. IEEE, ۲۰۱۷.
- Communication Technology for the Muslim World (ICTEM), pp. ۱۷۹-184. IEEE, ۲۰۱۸.
- [۴] Berk, Arslan, Rustam Olexandrovich Gamzayev, Ertuğrul Karaçuha, and Mykola Vyacheslavovich Tkachuk. "Algorithms and software solutions for SQL injection vulnerability testing in web applications." *Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології ۲۲ (۲۰۱۸): ۳-10.*
- [5] El Hayat, Soumiya Ain. "Intrusion Detection Systems: To an Optimal Hybrid Intrusion Detection System." In *Smart Data and Computational Intelligence: Proceedings of the International Conference on Advanced Information Technology, Services and Systems (AITSS-۱۸) Held on October ۱۷-۱۸, ۲۰۱۸ in Mohammedia*, p. ۲۸۴. Springer, ۲۰۱۹.
- [6] S. Khanna, and A. Verma, "Classification of SQL Injection Attacks Using Fuzzy Tainting", *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*. Springer, pp. ۴۶۳-469, ۲۰۱۸.
- [7] S. Steiner, D.Leon, and J. Alves-Foss, "A Structured Analysis of SQL Injection Runtime Mitigation Techniques", *Proceedings of the ۵۰th Hawaii International Conference on System Sciences*, ۲۰۱۷.
- [8] Liu, Guozhen. "SQL Injection Behavior Mining Based Deep Learning." In *Advanced Data Mining and Applications: ۱۴th International Conference, ADMA ۲۰۱۸, Nanjing, China, November ۱۶-۱۸, ۲۰۱۸, Proceedings*, vol. ۱۱۳۳۳, p. ۴۴۵. Springer, ۲۰۱۸.
- [9] Dewa, Zibusiso, and Leandros A. Maglaras. "Data mining and intrusion detection systems." *International Journal of Advanced Computer Science and Applications* ۷,۱ (۲۰۱۶): ۶۲-71.
- [10] M. Belhor, F Jemili, "Intrusion Detection Based on Genetic Fuzzy Classification System", ۹۷۸-1-5090-4320-0/۱۶, IEEE ۲۰۱۶.
- [11] S. Duquea, D. M. N. b. Omar, "Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS)", *Conference Organized by Missouri University of Science and Technology, San Jose, CA IEEE ۲۰۱۵.*
- [12] S. E. Benaicha, L. Saoudi, S. E. B. Guermeche, O. Lounis, "Intrusion Detection System Using Genetic Algorithm", *Science and Information Conference August ۲۷-29, London, UK IEEE ۲۰۱۴.*

- [25] Kar, Debabrata, Suvasini Panigrahi, and Srikanth Sundararajan. "SQLiDDS: SQL injection detection using query transformation and document similarity." In *International Conference on Distributed Computing and Internet Technology*, pp. 377-390. Springer, Cham, 2015.
- [22] Sukumar, JV Anand, I. Pranav, M. M. Neetish, and Jayasree Narayanan. "Network Intrusion Detection Using Improved Genetic k-means Algorithm." In 2018 *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2441-2446. IEEE, 2018.
- [23] Hanmanthu, B. B. Raghuram, and P. Niranjana. "SQL Injection Attack prevention based on decision tree classification." In 2015 *IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, pp. 1-5. IEEE, 2015.
- [24] Sahasrabuddhe, Atmaja, Sonali Naikade, Akshaya Ramaswamy, Burhan Sadliwala, and Pravin Futane. "Survey on intrusion detection system using data mining techniques." *Int Res J Eng Technol*, no. 5 (2017): 1780-4.
- [26] امینی، م. : "تشخیص تهاجم با استفاده از شبکه های عصبی"، پایان نامه کارشناسی ارشد، دانشگاه صنعتی شریف، 19-7، 1382