تشخیص نفوذ در شبکه‌های کامپیوتری به کمک الگوریتم سرمایش شبیه‌سازی شده

حمید محمدی، جعفر حبیبی، محمد صنیعی آباده، حمید سعدی

کارشناسی ارشد نرم افزاردانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف

hamid_m@ce.sharif.edu

دانشیار دانشکده مهندسی کامپیوتردانشگاه صنعتی شریف

jhabibi@sharif.edu

دانشجوی دکترا دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف

saniee@ce.sharif.edu

کارشناسی ارشد نرم افزاردانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف

saedi@ce.sharif.edu

۱- مقدمه

نفوذ[۱] به عنوان مجموعه‌ای از فعالیت‌ها تعریف می‌شود که تلاش می‌نمایند تا جامعیت، قابلیت اطمینان یا در دسترس بودن یک منبع را به خطر بیندازد [1]. سیستم تشخیص نفوذ[۲] (IDS) با استفاده از یک سری قوانین معین دسترسی کاربر را وارسی و محدود می‌نماید. این قوانین بر پایه دانش متخصص می‌باشد و از مدیران حرفه‌ای شبکه که سناریوی حملات را به منظور بررسی رفتار سیستم ایجاد نموده‌اند، حاصل شده است. سیستم تمام نفوذهای کاربران را مورد شناسایی قرار داده و فعالیت‌های لازم را برای متوقف نمودن حمله اتخاذ کرده و پیشنهاد می‌دهد. مساله تشخیص نفوذ به طور وسیعی در زمینه امنیت سیستم‌های کامپیوتری [5]-[2] مورد مطالعه قرار گرفته و اخیرا در زمینه‌های یادگیری ماشین و داده کاوی [8]-[6] مورد توجه بسیاری قرار گرفته است. تشخیص موارد سوء[۳] استفاده و تشخیص موارد ناهنجار[۴] دو رویکرد سیستم‌های تشخیص نفوذ می‌باشند [2]. رویکرد تشخیص موارد سوء استفاده، توانایی مشخص نمودن نفوذها را به کمک الگوهای شناخته شده از رفتارهای مخرب کسب می‌کند. سازندگان IDS که از این رویکرد استفاده می‌نمایند با به روز در آوردن این الگوها که با نام امضا شناخته می‌شوند، سیستم‌های کاربران خود را در مقابل آسیب پذیری‌های جدید محافظت می‌نمایند [11]-[9]. در رویکرد تشخیص موارد ناهنجاری، مدیر شبکه وضعیت عادی بار ترافیک شبکه را تعریف می‌نماید و با مشاهده رفتارهایی که از وضعیت عادی پیروی نمی‌نمایند موارد ناهنجار را تشخیص می‌دهد [14]-[12]. در این مقاله روش پیشنهادی از رویکرد تشخیص موارد سوء استفاده به منظور تشخیص نفوذ در شبکه‌های کامپیوتری استفاده می‌کند.

سیستم‌های فازی مبتنی بر قوانین اگر-آنگاه فازی به طور موفقیت آمیز در زمینه‌های مختلفی مورد استفاده قرار گرفته‌اند. قوانین اگر-آنگاه فازی به طور سنتی توسط افراد خبره بدست می‌آمد. اخیرا روش‌های متعددی به منظور ایجاد و تنظیم اتوماتیک قوانین اگر-آنگاه فازی بدون دخالت افراد خبره پیشنهاد شده است [15]. یکی از موارد چالش برانگیز در ایجاد سیستم‌های فازی اطمینان از واکشی اتوماتیک قوانین دسته‌بندی بهینه از مجموعه داده آموزشی می‌باشد، به طوری که قوانین بدست آمده دارای دقت بالا و قابلیت تفسیر خوبی برای افراد خبره باشد.

الگوریتم سرمایش شبیه‌سازی شده[۵] (SA) یک روش جستجوی تکراری و مکاشفه‌ای است که از حرارت و سرمایش فیزیکی فلزات برگرفته شده است و اولین بار توسط Metropolis در سال ۱۹۵۳ ارائه شد [16]. اولین تلاش‌ها به منظور وارد کردن SA در مسائل بهینه‌سازی توسط Kirkpatrick در سال ۱۹۸۳ صورت گرفت [17] که از SA به منظور فرار از راه حل‌های بهینه محلی و همگرایی

به سمت راه حل بهینه سراسری استفاده نمود. در این مقاله ما سیستم فازی مبتنی بر الگوریتم سرمایش شبیه‌سازی شده را برای ایجاد یک سیستم IDS با رویکرد تشخیص موارد سوء استفاده به کار گرفته‌ایم. روش پیشنهادی توسط مجموعه داده جامع تشخیص نفوذ KDD-Cup99 که به طور گسترده‌ای برای وارسی IDSها مورد استفاده قرار گرفته است، ارزیابی می‌شود. مجموعه داده KDD-Cup99 بر روی وب سایت دانشگاه کالیفرنیا-اروین موجود می‌باشد [18].

روش‌های متعددی برای مساله تشخیص نفوذ ارائه شده‌اند. شبکه‌های عصبی به طور گسترده‌ای برای تشخیص موارد سوء استفاده و ناهنجاری به کار رفته‌اند [19, 20]. تعدادی از کارهای انجام شده از سیستم‌های ایمنی مصنوعی برای تشخیص رفتارهای نفوذی بهره گرفته‌اند [21, 22]. تعدادی از تکنیک‌های کاربردی دیگر برای مساله تشخیص نفوذ، روش‌های مبتنی بر قانون فازی ژنتیکی [10,11,23]، تخمین پارامتریک بیزین [8] و خوشه‌بندی می‌باشند [24, 27].

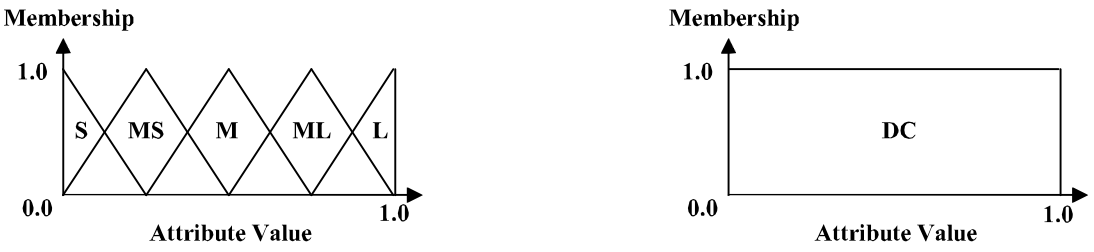
بخش‌های دوم و سوم مفاهیم پایه را ارائه می‌دهند. در بخش چهارم سیستم فازی مبتنی بر الگوریتم SA پیشنهادی را بیان می‌شود. در بخش پنجم نتایج به دست آمده و ارزیابی روش پیشنهادی با روش‌های معروف در این زمینه آمده است. نتیجه گیری در بخش ششم آمده است.

۲- مساله دسته‌بندی توسط قوانین اگر-آنگاه فازی

مساله دسته‌بندی الگو یک مساله با c دسته در فضای الگوی n-بعدی با مقادیر پیوسته می‌باشد. همچنین m بردار حقیقی به عنوان نمونه‌های آموزشی $x_p = (x_{p1}, x_{p2}, ..., x_{pn}), p = 1,2,...,m$ از c دسته داده شده‌اند. تمام مقادیر صفات نمونه‌ها در بازه یکتای $[0,1]$ نرمال شده‌اند. در سیستم دسته‌بندی فازی ارائه شده، از قوانین اگر-آنگاه فازی به صورت زیر استفاده می‌شود:

$$Rule\ R_j: \text{If } x_1 \text{ is } A_{j1} \text{ and } \dots \text{ and } x_n \text{ is } A_{jn}, \text{ then Class } C_j \text{ with } CF = CF_j .$$

که R_j برچسب jامین قانون اگر-آنگاه فازی می‌باشد، $A_{j1},...,A_{jn}$ مجموعه‌های فازی مقدم در بازه یکتا می‌باشند، C_j دسته نتیجه و CF_j درجه قطعیت مربوط به قانون می‌باشد. در پیاده‌سازی کامپیوتری از مجموعه‌های فازی مشابه شکل (۱) به عنوان مقادیر جزء مقدم قوانین استفاده می‌شود. تعداد کل قوانین اگر-آنگاه فازی در مساله دسته‌بندی الگوی n-بعدی برابر با 6^n می‌باشد. استفاده از همه این قوانین در یک پایگاه قانون فازی یکتا هنگامی که n بزرگ باشد غیر ممکن می‌باشد (برای مساله تشخیص نفوذ n برابر ۴۱ است).



شکل (۱): مجموعه‌های فازی استفاده شده به عنوان جزء مقدم قوانین:
1: Small, 2: Medium Small, 3: Medium, 4: Medium Large, Large, 6: Don't Care.

سیستم دسته‌بندی فازی ما به دنبال تعداد کمی قانون اگر-آنگاه فازی که دارای دقت بالایی در دسته‌بندی می‌باشند، می‌گردد. از آنجایی که دسته نتیجه و درجه قطعیت هر یک از قوانین اگر-آنگاه فازی از روی نمونه‌های آموزشی و توسط روال ابتکاری ساده‌ای به دست می‌آید [28] وظیفه سیستم دسته‌بندی ما ایجاد ترکیبی از مجموعه‌های فازی مقدم برای برای مجموعه قوانین اگر-آنگاه فازی می‌باشد. ممکن است این امر در وهله اول ساده به نظر رسد اما در حقیقت برای مسائل دسته‌بندی الگو با ابعاد بالا بسیار مشکل می‌باشد چون فضای حالت 6^n حالت دارد.

در سیستم دسته‌بندی فازی ما دسته نتیجه C_j و درجه قطعیت هر قانون CF_j از روال ابتکاری که در مرجع [28] معرفی شده است به صورت زیر محاسبه می‌شوند:

مرحله ۱- محاسبه درجه سازگاری هر نمونه آموزشی $x_p = (x_{p1}, x_{p2}, ..., x_{pn})$ با هر قانون اگر-آنگاه فازی R_j به کمک عملیات حاصل ضرب زیر:

$$\mu_{R_j}(x_p) = \mu_{A_{j1}}(x_{p1}) \times ... \times \mu_{A_{jn}}(x_{pn}), \quad p = 1, 2, ..., m \qquad (۱)$$

که $\mu_{A_{ji}}(.)$ تابع عضویت A_{ji} است.

مرحله ۲- محاسبه مجموع درجه‌های سازگاری برای هر دسته:

$$\beta_{Class\ h}(R_j) = \sum_{x_p \in Class\ h} \mu_{R_j}(x_p), \quad h = 1, 2, ..., c \qquad (۲)$$

مرحله ۳- یافتن دسته نتیجه C_j که بیشترین مقدار $\beta_{Class\ h}(R_j)$ را در بین c دارد.

$$\beta_{Class\ C_j}(R_j) = \max\{\beta_{Class\ 1}(R_j), ..., \beta_{Class\ c}(R_j)\}. \qquad (۳)$$

اگر بیش از یک دسته دارای بیشترین مقدار باشند آنگاه دسته نتیجه به صورت یکتا مشخص نمی‌شود و در این حالت یک مقدار تهی را به عنوان دسته نتیجه قانون در نظر می‌گیریم.

مرحله ۴- اگر دسته نتیجه C_j برابر با تهی بود آنگاه درجه قطعیت را برابر با صفر در نظر می‌گیریم $CF_j = 0$. در غیر این صورت درجه قطعیت به صورت زیر مشخص می‌شود:

$$CF_j = \frac{\beta_{Class\ C_j}(R_j) - \bar{\beta}}{\sum_{h=1}^{c} \beta_{Class\ h}(R_j)} \qquad (۴)$$

که $\bar{\beta}$ از رابطه زیر بدست می‌آید:

$$\bar{\beta} = \frac{\sum_{h \neq C_j} \beta_{Class\ h}(R_j)}{c - 1} \qquad (۵)$$

به کمک روال فوق برای هر قانون دسته نتیجه و درجه قطعیت هر قانون را مشخص می‌شود. برای یک الگوی جدید $x_p = (x_{p1}, ..., x_{pn})$ قانون برنده R_{j^*} به کمک رابطه زیر بدست می‌آید:

$$\mu_{j^*}(x_p).CF_{j^*} = \max\{\mu_j(x_p).CF_j : j = 1, 2, ..., N\} \qquad (۶)$$

برای الگوهای جدید دسته نتیجه و درجه قطعیت آن توسط قانون برنده‌ای که از رابطه (۶) برای الگو مشخص می‌شود، بدست می‌آیند.

روشی که برای کد کردن قوانین فازی در این مقاله استفاده شده است مشابه روش کد کردنی می‌باشد که در [11] استفاده نموده‌ایم. هر قانون اگر-آنگاه فازی به صورت یک رشته کد می‌شود. علائم زیر برای مشخص کردن پنج مقدار زبانی و بدون اهمیت: (شکل ۱) 1:small، 2:medium small، 3:medium، 4: medium large، 5:large و 6:don't care مورد استفاده قرار می‌گیرند. به عنوان مثال قانون فازی زیر به صورت "245163" کد می‌شود:

If x_1 is medium small and x_2 is medium large and x_3 is large and x_4 is small and x_5 is don't care and x_6 is medium, then Class C_j with $CF = CF_j$.

۳- الگوریتم سرمایش شبیه‌سازی شده

الگوریتم SA یک روش جستجوی تکراری است که از حرارت فیزیکی فلزات گرفته شده است [17]. این روش با یک راه حل اولیه آغاز می‌شود و با استفاده از آشفتگی‌ها و توابع ارزیابی مناسب به جستجوی تصادفی در فضای حالت می‌پردازد. راه حل‌های نامطلوب بر پایه

احتمالاتی که به کمک پارامتر دما (T) مشخص می‌شود، پذیرفته می‌شوند. احتمال پذیرش راه‌حل‌های نامطلوب در دماهای بالا، بیشتر است و با کاهش دما این احتمال کاهش می‌یابد. در دماهای بالا جستجو تقریبا تصادفی است، در حالی که در دماهای پایین جستجو تقریبا حریصانه می‌شود. در دمای صفر جستجو کاملا حریصانه است و فقط راه‌حل‌های خوب مورد قبول واقع می‌شوند.

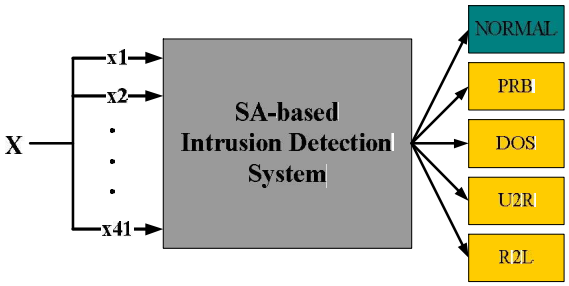
فرآیند این روش بدین شرح است؛ ابتدا یک شیء تا دمای بالایی حرارت داده می‌شود، سپس به آهستگی سرد می‌شود به طوری که سیستم تقریباً در هر زمان در تعادل ترمودینامیکی قرار داشته باشد. در حالت تعادل، شیء پیکربندی‌های زیادی دارد که با یک سطح انرژی مشخص شده متناظر است. از پیکربندی فعلی به پیکربندی بعدی یک آشفتگی تصادفی اعمال می‌شود تا سطح انرژی متناظر با آن، از سطح انرژی فعلی بدست آید. فرض کنید E_c و E_n سطح انرژی فعلی و سطح انرژی جدید را نمایش دهند. اگر $E_c > E_n$ آنگاه سطح انرژی جدید پذیرفته می‌شود، در غیر این صورت سطح انرژی جدید توسط تابع توزیع احتمالاتی بولتزمن به صورت زیر پذیرفته می‌شود:

$$p = \exp(-(E_n - E_c)/T) \qquad (۷)$$

که T دمای مربوط به آن تعادل می‌باشد. بنابراین احتمال پذیرفته شدن یک سطح انرژی بدتر هنگامی که T بالا باشد بیشتر و هنگامی که پایین باشد کمتر است. با کاهش تدریجی دما و تکرار شبیه‌سازی Metropolis، تا هنگامی که بهبود بیشتری ممکن نباشد، سطوح انرژی جدیدتری بدست خواهد آمد. دمای اولیه باید به دقت انتخاب شود زیرا اگر مقدار آن خیلی بالا باشد، زمان زیادی هدر می‌رود و اگر مقدار آن پایین باشد، ممکن است در جواب بهینه محلی گرفتار شویم.

۴- سیستم تشخیص نفوذ فازی مبتنی بر SA

در مساله تشخیص نفوذ نمونه‌های ورودی دارای ۴۱ صفت می‌باشند که این نمونه‌ها متعلق به پنج دسته می‌باشند. یک دسته عادی Normal و چهار دسته حمله PRB، DOS، U2R و R2L. شکل (۲) سیستم پیشنهادی دسته‌بندی فازی مبتنی بر SA را برای مساله تشخیص نفوذ نشان می‌دهد:



شکل (۲): سیستم تشخیص نفوذ فازی مبتنی بر SA

طرح کلی الگوریتم پیشنهادی به صورت زیر می‌باشد:

گام ۱- ایجاد مجموعه اولیه قوانین اگر-آنگاه فازی و مقداردهی دمای سیستم به دمای اولیه.

گام ۲- ارزیابی مجموعه قوانین اگر-آنگاه فازی فعلی به کمک تابع هزینه ($COST$).

گام ۳- ایجاد مجموعه قوانین جدید از مجموعه قوانین فعلی به کمک توابع آشفتگی.

گام ۴- پذیرش مجموعه قوانین جدید به شرطی که $COST_{new} < COST_{current}$ در غیر این صورت مجموعه قوانین جدید با استفاده از تابع توزیع احتمال بولتزمن که در رابطه (۷) آمده است مورد پذیرش قرار خواهد گرفت.

گام ۵- تکرار گام‌های ۲-۴ به اندازه k بار در هر دما.

گام ۶- کاهش دما توسط ضریب سرد شدن.

گام ۷- خاتمه الگوریتم در صورت برآورده شدن شرط خاتمه. در غیر این صورت به گام ۲ بر می‌گردیم.

گام ۱- مقداردهی اولیه:

فرض کنید N_{init} تعداد قوانین فازی را در مجموعه قوانین اولیه نشان می‌دهد. در این مقاله برای ایجاد مجموعه قوانین اولیه، تعداد N_{init} قانون اگر-آنگاه فازی به صورت تصادفی و با مقداردهی جزء اول قوانین به کمک مقادیر زبانی موجود در شکل (۱) تولید می‌شوند. در اینجا ما احتمال نسبت دادن *don't care* را بالاتر از پنج مقدار دیگر در نظر گرفته‌ایم تا بدین ترتیب احتمال به دست آمدن قوانین کلی‌تر را افزایش دهیم. بعد از ایجاد N_{init} قانون اگر-آنگاه فازی به عنوان مجموعه قوانین اولیه S_{init}، دسته نتیجه و درجه قطعیت هر قانون با استفاده از روال ابتکاری که در بخش‌های پیشین مطرح شد از مجموعه داده آموزشی به دست می‌آید. شایستگی هر قانون از رابطه زیر حاصل می‌شود:

$$fitness(R_j) = NCP(R_j) \quad (۸)$$

که $NCP(R_j)$ تعداد نمونه‌هایی از مجموعه داده آموزشی می‌باشد که به درستی توسط قانون R_j دسته‌بندی شده‌اند.

گام ۲- ارزیابی:

مجموعه قوانین اگر-آنگاه فازی باید دارای درجه قطعیت بالایی باشد. پس ما از تابع هزینه زیر برای ارزیابی مجموعه قوانین استفاده می‌کنیم:

$$COST(S) = m - \sum_{j=1}^{N} NCP(R_j) \quad (۹)$$

که m تعداد کل نمونه‌های مجموعه آموزشی است و $\sum_{j=1}^{N} NCP(R_j)$ تعداد نمونه‌هایی است که به درستی توسط مجموعه قوانین S دسته‌بندی شده اند و N تعداد قوانین اگر-آنگاه فازی موجود در مجموعه قوانین S می‌باشد.

گام ۳- آشفتگی:

فرایند آشفته‌سازی مجموعه قوانین فعلی برای به دست آوردن مجموعه قوانین جدید به کمک دو تابع صورت می‌گیرد که عبارتند از: اصلاح یک قانون در مجموعه قوانین (*Modify*) و حذف یک قانون از مجموعه قوانین (*Delete*):

- *Mofidy*: یک قانون به صورت تصادفی انتخاب شده سپس یکی از مقادیر زبانی جزء مقدم آن به مقادیر دیگر تغییر می‌کند. در صورتی که دسته نتیجه قانون اصلاح شده با دسته نتیجه قانون انتخاب شده برابر باشد، آنگاه قانون اصلاحی جایگزین قانون انتخاب شده می‌گردد وگرنه این روال دوباره ادامه می‌یابد.
- *Delete*: یک قانون از بین مجموعه قوانین فعلی با استفاده از رابطه احتمالاتی زیر انتخاب شده و حذف می‌گردد:

$$P(R) = \frac{fitness_{\max}(S_{Class\ h}) - fitness(R)}{fitness_{\max}(S_{Class\ h}) - fitness_{\min}(S_{Class\ h})} \quad (۱۰)$$

که در رابطه فوق $fitness_{\max}(S_{Class\ h})$ و $fitness_{\min}(S_{Class\ h})$ به ترتیب بیشترین و کمترین مقدار شایستگی قوانین انتخاب شده از دسته انتخاب شده، $S_{Class\ h}$ می‌باشد. در اینجا قوانینی که دارای مقدار شایستگی کمتری می‌باشند با احتمال بالاتری انتخاب می‌شوند.

گام ۴- پذیرش: اگر مقدار تابع هزینه مجموعه قوانین جدید $COST_{new}$ کمتر از مقدار تابع هزینه بهترین مجموعه قوانین $COST_{best}$ باشد، آنگاه مجموعه قوانین جدید مورد پذیرش قرار گرفته و جایگزین مجموعه قوانین فعلی و بهترین مجموعه قوانین می‌شود. اگر مقدار تابع هزینه مجموعه قوانین جدید فقط از مجموعه قوانین فعلی کمتر باشد جایگزین این مجموعه قوانین شده و در نهایت اگر مقدار هزینه مجموعه قوانین جدید بیشتر از مجموعه قوانین فعلی باشد آنگاه Metropolis با استفاده از تابع چگالی احتمال بولتزمن در مورد پذیرش مجموعه قوانین جدید تصمیم‌گیری خواهد کرد.

گام ۵ و۶ و ۷ – تکرار، کاهش دما و خاتمه: در هر دما روال Metropolis به تعداد k بار تکرار می‌شود تا فضای حالت به خوبی در هر دما مورد جستجو قرار گیرد. ضریب سرد شدن α، به منظور کاهش دما مورد استفاده قرار می‌گیرد. دما باید به آهستگی کاهش داده شود تا الگوریتم به خوبی بتواند در دماهای بالا فضای مساله را مورد explore قرار دهد و در دماهای پایین به سمت جواب بهینه

سراسری exploit نماید. هنگامی که دما به دمای کمینه $T_{\min}$ رسید الگوریتم خاتمه می‌یابد و بهترین مجموعه قوانین به عنوان خروجی برگردانده می‌شود.

۵- نتایج بدست آمده و ارزیابی روش پیشنهادی

روش پیشنهادی بر روی مجموعه داده KDD-Cup99 اجرا شده است. جدول (۱) توزیع بخشی از نمونه‌های این مجموعه داده را که ۱۰ درصد کل مجموعه داده را در بر دارد نشان می‌دهد.

جدول (۱): توزیع نمونه‌ها در ۱۰ درصد KDD-Cup99

نمونه‌ها	زیر-دسته	دسته
۹۷۲۷۸		NORMAL
۴۱۰۷	ipsweep, nmap, portsweep, satan	PRB
۳۹۱۴۵۸	back, land, Neptune, pod, smurf, teardrop	DOS
۵۲	buffer_overflow, loadmodule, multihop, perl, rootkit	U2R
۱۱۲۶	ftp_write, guess_passwd, imap, phf, spy, warezclient, warezmaster	R2L

همانطور که در جدول (۱) ملاحظه می‌شود تعداد نمونه‌های موجود در ۱۰ درصد مجموعه داده بسیار زیاد می‌باشد (۴۹۴۰۲۱ نمونه). در اینجا ما از زیرمجموعه‌ای با تعداد ۲۰۷۵۲ نمونه از تمام دسته‌ها به عنوان مجموعه داده آموزشی[۶] و ۳۱۱۰۲۹ نمونه به عنوان مجموعه داده آزمایشی[۷] استفاده نموده‌ایم. توزیع نمونه‌های این دو مجموعه در جدول (۲) ارائه شده است.

جدول (۲): توزیع نمونه‌ها در مجموعه داده آموزشی و آزمایشی

دسته	آموزشی	آزمایشی
NORMAL	۱۰۰۰۰	۶۰۵۹۳
PRB	۴۱۰۷	۴۱۶۶
DOS	۵۴۶۷	۲۲۹۸۵۳
U2R	۵۲	۲۲۸
R2L	۱۱۲۶	۱۶۱۸۹

الگوریتم پیشنهادی روش پیشنهادی را با پارامترهای موجود در جدول (۳) برای مساله تشخیص نفوذ به کارمی‌بریم.

جدول (۳): مقداردهی پارامترهای روش پیشنهادی

پارامتر	مقدار
تعداد قوانین مجموعه اولیه (N_{init})	۸۰
دمای اولیه ($T_{\max}$)	۱۰۰
دمای نهایی ($T_{\min}$)	۰.۰۰۰۱
ضریب کاهش دما (α)	۰.۹۵
تعداد تکرارها در هر دما (k)	۸۰

میانگین دقت دسته‌بندی از ۹۶ درصد تا ۹۹ درصد می‌باشد. مجموعه قوانین با ۶۸ قانون و دقت دسته‌بندی ۹۶.۵۷ برای مرحله آزمایش مورد استفاده قرار گرفته است. جدول (۴) ماتریس پراکندگی روش پیشنهادی را نمایش می‌دهد.

جدول (۴): ماتریس پراکندگی روش پیشنهادی

	دسته تشخیص داده شده					
دسته واقعی	NORMAL	PRB	DOS	U2R	R2L	%
NORMAL	۵۹۸۵۹	۲۰۳	۸۹	۲۴	۳۷۹	۹۸.۷۸
PRB	۶۴۷	۳۱۹۴	۳۰۶	.	.	۷۶.۶۶
DOS	۲۴۸۱	۵۷۶	۲۲۶۳۸۸	۱۰	۰	۹۸.۴۹
U2R	۶۳	۱۱۶	۰	۳۷	۱۱	۱۶.۲۲
R2L	۱۰۸۶۳	۷۷	۳۴۴۷	۸	۱۹۷۱	۱۲.۱۷
%	۸۱.۱۸	۷۶.۶۶	۹۸.۳۳	۴۶.۸۳	۸۳.۴۸	۹۳.۷۰

همانطور که در جدول (۴) ملاحظه می‌شود، تعداد ۵۹۸۵۹ نمونه از ۶۰۵۹۳ نمونه Normal بـه درسـتی توسـط روش پیشـنهادی دسته‌بندی شده‌اند. درصد موجود در ستون آخر نشان می‌دهد که ۹۸.۷۸ درصد از نمونه‌های Normal به درستی دسته‌بندی شـده‌انـد. دقت‌های ارائه شده در سطر آخر نشان می‌دهد که ۸۱.۱۸ درصد از نمونه‌هایی که Normal تشخیص داده شـده‌انـد، حقیقتـا Normal بوده‌اند. مشابه همین استنتاج برای دسته‌های دیگر وجود دارد. با توجه به سطر و سـتون آخـر دقـت نهـایی روش پیشـنهادی برابـر بـا ۹۳.۷۰ درصد می‌باشد. جدول (۵) ماتریس هزینه دسته‌بندی اشتباه را نشان می‌دهد [29]:

جدول (۵): ماتریس هزینه دسته‌بندی اشتباه [29]

	دسته تشخیص داده شده				
دسته واقعی	NORMAL	PRB	DOS	U2R	R2L
NORMAL	۰	۱	۲	۲	۲
PRB	۱	۰	۲	۲	۲
DOS	۲	۱	۰	۲	۲
U2R	۳	۲	۲	۰	۲
R2L	۴	۲	۲	۲	۰

جدول (۶) هزینه دسته‌بندی اشتباه روش پیشنهادی را ارائه می‌دهد. درایه موجود در سطر و ستون آخر نشان می‌دهـد کـه هزینـه دسته‌بندی روش پیشنهادی برابر با ۰.۱۸۷۲ می‌باشد.

جدول (۶): هزینه دسته‌بندی اشتباه روش پیشنهادی

	دسته تشخیص داده شده					
دسته واقعی	NORMAL	PRB	DOS	U2R	R2L	
NORMAL	۰	۲۰۳	۱۷۸	۴۸	۷۵۸	
PRB	۶۴۷	۰	۶۱۲	۰	۰	
DOS	۴۹۶۲	۵۷۶	۰	۲۰	۰	
U2R	۱۸۹	۲۳۲	۰	۰	۲۲	
R2L	۴۲۷۳۲	۱۵۴	۶۸۹۴	۱۶	۰	
						۰.۱۸۷۲

مقایسه روش پیشنهادی با روش‌های دیگر

کارایی دسته‌بندی روش پیشنهادی روش پیشنهادی با روش‌های C4.5، Naïve Bayes (NB)، K- نزدیکترین همسایه (K-NN)، ماشین بردار پشتیبان (SVM)، سیستم تشخیص نفوذ فازی ژنتیکی چند منظوره (MOGF-IDS) و Winner Entryمورد مقایسه قرار گرفته است. در روش K-NN مقدار K برابر با پنج انتخاب شده است. جدول (۷) دقت نهایی و هزینه دسته‌بندی را ارائه می‌دهد و جدول (۸) مقادیر Recall، Precision و F-Measure روش روش پیشنهادی را برای روش‌های بالا نشان می دهد.

با توجه به جدول (۷) روش پیشنهادی روش پیشنهادی بالاترین دقت کل دسته‌بندی و همچنین کمترین هزینه را برای مساله تشخیص نفوذ اخذ نموده است. این امر به این دلیل است که روش پیشنهادی با مقداردهی اولیه مناسب، توابع آشفتگی مفید و تابع ارزیابی دقیق به خوبی در فضای حالت مساله از راه حل‌های بهینه محلی گریخته و به سمت راه حل بهینه سراسری حرکت می‌نماید.

جدول (۸) نشان می‌دهد که روش پیشنهادی برای نمونه‌های Normal دارای بهترین دقت Precision و F-Measure می‌باشد و به همراه روش Winner Entry بالاترین دقت‌ها را برای Recall دارند. برای دسته حملات PRB روش پیشنهادی دارای بهترین دقت Precision و F-Measure می‌باشد. برای دسته حملات بسیار شایع DOS الگوریتم ما بهترین دقت Recall و F-Measure را به دست آورده است. برای حملات U2R روش پیشنهادی، بهترین دقت Recall و سومین دقت F-Measure را کسب نموده است. برای آخرین دسته حملات، R2L، روش پیشنهادی دومین دقت Precision و سومین دقت Recall و F-Measure را کسب نموده است. با توجه به نتایج به دست آمده ملاحظه می‌شود که روش پیشنهادی با تشخیص حملات با نرخ تشخیص بالا و نمونه‌های عادی با نرخ هشدارهای اشتباه پایین کارایی قابل ملاحظه‌ای دارد.

جدول (۷): دقت نهایی و هزینه دسته‌بندی روش‌های مختلف

روش	دقت	هزینه دسته‌بندی
C4.5 [23]	۹۲.۰۴	۰.۲۴۸۰
NB [23]	۷۶.۴۵	۰.۴۹۶۵
5-NN [23]	۹۱.۸۳	۰.۲۴۵۸
SVM [23]	۹۲.۵۴	۰.۲۴۵۷
MOGF-IDS [23]	**۹۲.۷۷**	**۰.۲۳۱۷**
Winner Entry[29]	۹۲.۷۱	۰.۲۳۳۱
روش پیشنهادی	**۹۳.۷۰**	**۰.۱۸۷۲**

جدول (۸): Recall، Precision و F-Measure روش‌های مختلف. بهترین مقادیر با قلم درشت و خط زیر، دومین مقادیر با قلم درشت و سومین مقادیر با خط زیر مشخص شده‌اند.

دسته	روش	C4.5 [23]	NB [23]	5-NN [23]	SVM [23]	MOGF-IDS[23]	Winner Entry [29]	روش پیشنهادی
NORMAL	Recall	۹۸.۳۸	۵۵.۴۷	۹۵.۸۹	۹۷.۹۹	۹۸.۳۶	**۹۹.۵۰**	**۹۸.۷۸**
	Precision	**۷۴.۷۵**	۴۳.۳۳	۷۴.۱۵	۷۳.۴۲	۷۴.۷۴	۷۴.۶۱	**۸۱.۱۸**
	F-measure	۸۴.۹۶	۴۸.۶۵	۸۳.۶۳	۸۳.۹۴	۸۴.۹۴	**۸۵.۲۸**	**۸۹.۱۲**
PRB	Recall	۸۱.۸۸	**۹۰.۴۵**	۸۱.۶۱	۸۶.۲۷	**۸۸.۶۰**	۸۳.۳۰	۷۶.۶۶
	Precision	۵۲.۲۰	۶۴.۱۶	۵۵.۴۶	**۷۷.۷۲**	۷۴.۴۰	۶۴.۸۱	**۷۶.۶۶**
	F-measure	۶۳.۷۶	۷۵.۰۷	۶۶.۰۵	**۸۱.۷۷**	**۸۰.۸۸**	۷۲.۹۰	۷۶.۶۶
DOS	Recall	۹۶.۹۹	۸۲.۷۵	۹۷.۰۰	**۹۷.۶۵**	۹۷.۲۰	۹۷.۱۰	**۹۸.۴۹**
	Precision	۹۹.۶۹	۹۴.۰۰	۹۹.۴۲	۹۹.۸۶	**۹۹.۹۰**	**۹۹.۸۸**	۹۸.۳۳
	F-measure	۹۸.۳۲	۸۸.۰۲	۹۸.۱۹	**۹۸.۷۰**	**۹۸.۵۳**	۹۸.۴۷	۹۸.۴۱
U2R	Recall	۱۴.۴۷	۱۳.۱۶	۱۴.۹۱	۱۰.۰۹	**۱۵.۷۹**	۱۳.۲۰	**۱۶.۲۲**
	Precision	۹.۳۵	۲.۰۵	۵.۴۷	۵۳.۴۹	**۶۱.۰۲**	**۷۱.۴۳**	۴۶.۸۳
	F-measure	۱۱.۳۶	۳.۵۴	۸.۰۰	۱۶.۹۷	**۲۵.۰۹**	۲۲.۲۸	**۲۴.۱۰**

	Recall	۱.۴۵	**۶۲.۷۴**	۶.۹۰	۳.۵۵	۱۱.۰۱	۸.۴۰	**۱۲.۱۷**
R2L	Precision	۳۰.۳۲	۴۲.۷۰	۶۶.۹۷	۶۲.۳۹	۶۸.۳۹	**۹۸.۸۴**	**۸۳.۴۸**
	F-measure	۲.۷۷	**۵۰.۸۲**	۱۲.۵۱	۶.۷۱	۱۸.۹۷	۱۵.۴۸	**۲۱.۲۵**

۶- نتیجه

در این مقاله روشی را به منظور تشخیص رفتارهای نفوذی در شبکه‌های کامپیوتری معرفی نمودیم. نتایج به دست آمده بر روی مجموعه داده KDD-Cup99 نشان داد که روش روش پیشنهادی دارای کارایی قابل ملاحظه‌ای در تشخیص حملات نفوذی و ترافیک عادی شبکه‌ها می‌باشد.

از آنجایی که مساله تشخیص نفوذ یک مساله با ابعاد بالایی می‌باشد، روش پیشنهادی با مقداردهی‌های اولیه، توابع آشفتگی و تابع هزینه مناسب در فضای بسیار پیچیده مساله از راه حل‌های بهینه محلی گریخته و به سمت راه حل بهینه سراسری حرکت می‌نماید.

برای ادامه کار می‌توان معیار قابلیت تفسیر مربوط به قوانین فازی را به عنوان یکی از معیارهای تابع ارزیابی در نظر گرفت تا علاوه بر دقت خوب، خروجی دارای قابلیت تفسیر بالایی باشد که در آینده به آن خواهیم پرداخت.

تشکر و قدردانی

این تحقیق با حمایت مرکز تحقیقات مخابرات ایران انجام پذیرفته است.

مراجع

[1] R. Heady, G. Luger, A. Maccabe, and M. Servilla, “The architecture of network level intrusion detection system,” Technical Report, Department of Computer Science, University of New Mexico, 1990.
[2] A. Murali, M. Rao, “A Survey on Intrusion Detection Approaches,” First International Conference on Information and Communication Technologies, pp. 233- 240, 2005.
[3] N.B. Idris, B. Shanmugam, “Artificial Intelligence Techniques Applied to Intrusion Detection,” Annual IEEE INDICON, pp. 52-55, 2005.
[4] N. Ye, S. Vilbert, and Q. Chen, “Computer Intrusion Detection Through EWMA for Autocorrelated and Uncorrelated Data,” IEEE Transactions on Reliability, vol. 52, no. 1, pp. 75-82, Mar. 2003.
[5] N. Ye, Q. Chen, and C.M. Borror, “EWMA Forecast of Normal System Activity for Computer Intrusion Detection,” IEEE Transactions on Reliability, vol. 53, no. 4, pp. 557-566, Dec. 2004.
[6] S.B. Cho, “Incorporating soft computing techniques into a probabilistic intrusion detection system,” IEEE Transactions on Systems, Man and Cybernetics, Part C, Volume 32, Issue 2, pp.154-160, May 2002.
[7] J. Tian, Y. Fu, Y. Xu, J. Wang, “Intrusion Detection Combining Multiple Decision Trees by Fuzzy Logic,” Sixth International Conference on Parallel and Distributed Computing Applications and Technologies, pp. 256-258, 2005.
[8] S. Cho, S. Cha, “SAD: web session anomaly detection based on parameter estimation,” Computers & Security, Vol.23, No.4, pp.265-351, Jun. 2004.
[9] H.H. Gao, H.H. Yang, X.Y. Wang, “Ant Colony Optimization Based Network Intrusion Feature Selection and Detection,” Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, pp. 3871-3875, Aug. 2005.
[10] T. Ozyer, R. Alhajj, K. Barker, “Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening,” Journal of Network and Computer Applications, pp. 99-113, 2007.
[11] M.S. Abadeh, J. Habibi, and C. Lucas, “Intrusion Detection Using a Fuzzy Genetics-Based Learning Algorithm,” Journal of Network and Computer Applications, pp. 414-428, 2007.
[12] C. Kruegel and G. Vigna, “Anomaly Detection of Web-Based Attacks,” Proc. 10th ACM Conf. Computer and Comm. Security (CCS ’03), pp. 251-261, Oct. 2003.
[13] Y. Feng, Z.F. Wu, K.G. Wu, Z.Y. Xiong, Y. Zhou, “An Unsupervised Anomaly Intrusion Detection Algorithm Based On Swarm Intelligence,” Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, pp. 3965-3969, August 2005.
[14] A.A.E. Ahmed, and I. Traore, “Anomaly Intrusion Detection based on Biometrics,” Proceedings of the 2005 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, pp. 452-453, 2005.
[15] C.C. Lee, “Fuzzy logic in control systems: fuzzy logic controller, Part I and Part II,” IEEE Transactions on Systems, Man, and Cybernetics, 20(2), pp. 404–435, 1990.
[16] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, “Equation of state calculation by fast computing machines,” Journal of Chemical Physics, vol. 21, pp. 1087-1092, 1953.
[17] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, “Optimization by simulated annealing,” Science, vol. 220, pp. 671-680, 1983.

[18] KDD-Cup data set: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[19] H. Debar, B. Becke, and D. Siboni, "A neural network component for an intrusion detection system," In: Proceedings of the IEEE Computer Society symposium on research in security and privacy, pp. 240-250, 1992.
[20] S. Mukkamala, A.H. Sung, "Feature selection for intrusion detection using neural networks and support vector machines," Journal of the Transport Research Board National Academy, Transport Research Record No. 1822, pp. 33-39, 2003.
[21] D. Dasgupta, and F. González, "An Immunity-Based Technique to Characterize Intrusions in Computer Networks," IEEE Transactions on Evolutionary Computation, vol. 6, no. 3, pp. 1081-1088, Jun. 2002.
[22] P.K. Harmer, P.D. Williams, G.H. Gunsch, and G.B. Lamont, "An Artificial Immune System Architecture for Computer Security Applications," IEEE Transactions on Evolutionary Computation, vol. 6, no. 3, pp. 252-280, Jun. 2002.
[23] C.H. Tsang, S. Kwong, H. Wang, "Anomaly Intrusion Detection Using Multi-Objective Genetic Fuzzy System and Agent-Based Evolutionary Computation Framework," ICDM05, pp. 789-792, 2005.
[24] B. Xu, A. Zhang, "Application of Support Vector Clustering Algorithm to Network Intrusion Detection," ICNN&B '05: International Conference on Neural Networks and Brain, Volume 2, pp. 1036-1040, 2005.
[25] S.H. Oh, and W.S. Lee, "An anomaly intrusion detection method by clustering normal user behavior," Computers & Security, Vol. 22, No.7, pp. 596-612, Nov. 2003.
[26] E. Leon, O. Nasraoui, and J. Gomez, "Anomaly detection based on unsupervised niche clustering with application to network intrusion detection," in Proceedings of IEEE Conference on Evolutionary Computation (CEC), pp. 502-508, 2004.
[27] Y. Guan, A. A. Ghorbani, and N. Belacel, "Y-MEANS: a clustering method for intrusion detection," in Canadian Conference on Electrical and Computer Engineering, pp. 1083-1086, 2003.
[28] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," Fuzzy Sets and Systems, 52(1), pp. 21-32, 1992.
[29] C. Elkan, "Results of the KDD'99 classifier learning," ACM SIGKDD Explorations 1, pp. 63–64, 2000.

زیرنویس‌ها

[1] Intrusion
[2] Intrusion Detection System
[3] Misuse
[4] Anomaly
[5] Simulated annealing
[6] Train Set
[7] Test Set