تشخیص نفوذ به‌کمک سیستم‌های چندعامله مبتنی بر اتوماتاهای یادگیر

فرناز ابطحی
آزمایشگاه محاسبات نرم
دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
تهران ایران
abtahi@aut.ac.ir

محمدرضا میبدی
آزمایشگاه محاسبات نرم
دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
تهران ایران
mmeybodi@aut.ac.ir

چکیده: در این مقاله، یک مدل سلسله مراتبی از عامل‌های مبتنی بر اتوماتای یادگیر برای تشخیص نفوذهای از نوع انکار سرویس پیشنهاد میگردد. در مدل پیشنهادی، هر عامل دارای یک اتوماتای یادگیر است که پارامترهای مدل را نگهداری کرده و آن‌ها را با توجه به بازخوردهایی که از محیط دریافت می‌کنند به‌روز می‌رسانند. در مدل ارائه شده، مسئله تشخیص نفوذ به‌صورت یک مسئله دسته‌بندی دوکلاسه مطرح می‌گردد. برای آموزش و آزمایش این مدل، از مجموعه داده‌های *KDD99* استفاده شده که یک مجموعه داده استاندارد برای کاربردهای امنیتی می‌باشد. آزمایش‌ها نشان می‌دهند که سیستم تشخیص نفوذ پیشنهادی، در مقایسه با سیستم‌های مشابه، از عملکرد مناسبی از لحاظ سرعت، نرخ تشخیص و نرخ خطای مثبت نادرست بر خوردار است.

واژه هـای کلیـدی: تشـخیص نفـوذ، حمـلات *DoS*، سیسـتم‌هـای چندعامله، اتوماتاهای یادگیر

۱- مقدمه

حملات انکار سرویس[1] *(DoS)*، یکی از رایج‌ترین انواع حملات در سیستم‌های کامپیوتری می‌باشند. هدف این نوع حملات، ایجاد اختلال در روال سرویس‌دهی میزبان و یا شبکه است. برخی از این نوع حملات، سرویس‌های مجاز شبکه را بیش از حد فراخوانی می‌کنند، برخی دیگر بسته‌هایی دستکاری شده برای قربانی (هدف حمله) ارسال می‌کنند که او را دچار مشکل سازند و برخی از مشکلات نرم‌افزارهای شبکه بهره می‌برند. [1]

هدف از این مقاله طراحی مدلی برای تشخیص حملات ازکار انداختن سرویس (نفوذهای *DoS*) به کمک عامل‌های مبتنی بر اتوماتاهای یادگیر[2] است. مدل پیشنهادی در این مقاله که شامل سلسله‌مراتبی از عامل‌های مبتنی بر اتوماتای یادگیر می‌باشد، قادر است ترافیک شبکه را به‌عنوان حمله و یا نرمال دسته‌بندی نماید. در این سلسله‌مراتب، داده‌ها از بالاترین سطح وارد شده و تا پایین‌ترین سطح حرکت می‌کنند. در هر سطح، رکورد مورد نظر با توجه به وظیفه نسبت داده‌شده به همان سطح، پردازش شده و به سطح بعدی هدایت می‌گردد. در انتها، به هر رکورد از داده‌های ورودی، برچسب حمله یا نرمال نسبت داده می‌شود.

آزمایشات گوناگونی به منظور ارزیابی کارایی روش پیشنهادی در حوزه تشخیص نفوذ انجام گرفته است که نتایج آن‌ها در این مقاله ارائه و بررسی می‌گردد. اولین گام برای ارزیابی یک سیستم تشخیص نفوذ، انتخاب یک بستر یا مجموعه داده معتبر جهت انجام آزمایشات مربوطه می‌باشد. مجموعه داده *KDD99* [2] یکی از مرسوم‌ترین بسترهای ارزیابی سیستم‌های تشخیص نفوذ محسوب می‌شود. گام دیگر، تعریف معیارهای ارزیابی می‌باشد. دو معیار ارزیابی مرسوم برای سیستم‌های تشخیص نفوذ عبارتند از نرخ تشخیص و خطای مثبت نادرست (هشدار غلط) [3].

ادامه مقاله بدین صورت سازمان‌دهی شده‌است: در بخش2، مروری بر مجموعه داده‌های *KDD99* خواهیم داشت. در بخش3، معیارهای ارزیابی کارایی، یعنی نرخ تشخیص[3] و خطای مثبت نادرست[4] را معرفی خواهیم کرد. بخش4 شامل معرفی اتوماتای یادگیر و عملکرد آن می‌باشد. بخش5 به ارائه مدل پیشنهادی برای تشخیص نفوذ اختصاص خواهد داشت. بخش6 نتیجه گیری میباشدکه به ارایه نتایج آزمایشات و نکات قوت مدل پیشنهادی میپردازد.

۲- مجموعه داده‌های *KDD99*

گروه *IST* از آزمایشگاه *MIT Lincoln* زیر نظر *DARPA* و *AFRL/SNHS*، اولین داده‌های استاندارد برای بررسی و ارزیابی سیستم‌های تشخیص نفوذ را جمع‌آوری نمودند. این اطلاعات در طول چند هفته در یک شبیه‌سازی برای آزمایش سیستم تشخیص نفوذ *DARPA* به کار رفته‌اند. این مجموعه داده‌ها براساس سال جمع‌آوری اطلاعات (1998-1999) طبقه بندی شده است.

---

[1] Denial of Service
[2] Learning Automata
[3] Detection Rate
[4] False Positive Rate

مجموعه داده‌های سال 1999 که به اهتمام و با نظارت لی و در طی انجام پروژه دکترای وی جمع‌آوری شد [4]، در سومین مسابقه بین‌المللی اکتشاف دانش و داده‌کاوی، *KDD CUP 99* و در پنجمین کنفرانس در این زمینه مورد استفاده قرار گرفت. این بانک اطلاعاتی شامل رکوردهای اتصال استانداردی می‌باشد که مجموعه‌ای از حملات و نفوذهای شبیه‌سازی‌شده در یک شبکه نظامی را شامل می‌شود.

یک اتصال، دنباله‌ای از بسته‌های با پروتکل *TCP*، *UDP* و یا *ICMP* است که در زمان‌های مشخص شروع و پایان می‌یابد و بین آن زمتن‌ها، داده‌ها از آدرس *IP* مبدأ به آدرس *IP* مقصد و برعکس تحت یک قرارداد تعریف شده جریان دارند. هر اتصال به‌عنوان نرمال یا حمله برچسب‌گذاری می‌شود و در مورد حمله، نوع آن دقیقاً مشخص می‌گردد. رکورد هر اتصال شامل حدود صد بایت است.

حملاتی که در این مجموعه داده مشاهده می‌شوند، در چهار دسته اصلی *DOS* ، *R2L*، *U2R* و *Probe* قرار می‌گیرند. مجموعه داده‌های *KDD99* در قالب چند فایل سازماندهی گردیده‌اند. در این مجموعه از فایل‌ها، دو فایل به نام‌های *10_percent* و *corrected* وجود دارند که در بسیاری از پژوهش‌ها مانند [5] و [6]، به‌ترتیب به عنوان بسترهایی برای آموزش و آزمایش سیستم‌های طراحی‌شده به‌کار می‌روند.

جدول 1، تعداد رکوردهای موجود در هر یک از این فایل‌ها را به تفکیک نوع حمله (یکی از چهار دسته فوق) و نرمال نشان می‌دهد. هر رکورد از مجموعه داده *KDD99* ، یک اتصال را در قالب 41 ویژگی توصیف می‌کند که در این مجموعه از ویژگی‌ها، 32 ویژگی پیوسته و 9 ویژگی گسسته (نمادین) وجود دارد.

جدول 1- تعداد رکوردهاي مجموعه‌هاي *10_percent* و *corrected* به تفکیک نوع حمله/نرمال [2]

مجموعه داده	نرمال	DoS	U2R	R2L	Probe	تعداد کل
10_percent	97،278	391،458	52	1،126	4،107	494،021
corrected	60،593	229،853	70	16،347	4،166	311،029

۳- روش ارزیابی

کارایی سیستم‌های تشخیص نفوذ و حتی در حالت کلی‌تر کارایی دسته‌بندی‌کننده‌ها، غالباً از طریق دو معیار نرخ تشخیص و نرخ خطای مثبت نادرست ارزیابی می‌شود [7]. برای ارائه تعریف این دو معیار ابتدا باید به تعریف دو نوع خطای ممکن در سیستم‌های تشخیص نفوذ بپردازیم. این دو نوع عبارتند از:

- خطای مثبت نادرست: اگر یک اتصال نرمال به عنوان حمله شناسایی شود، یک خطای مثبت نادرست رخ داده است.
- خطای منفی نادرست[5]: اگر یک اتصال با برچسب حمله به عنوان یک اتصال نرمال شناسایی شود، یک خطای منفی نادرست رخ داده است.

با توجه به تعاریف فوق داریم:

$$\text{False Negative Rate} = \frac{\text{Number of False Negatives}}{\text{Total Number of Attack Conections}} \quad (1)$$

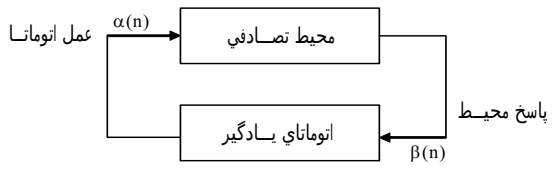
$$\text{False Positive Rate} = \frac{\text{Number of False Positives}}{\text{Total Number of Attack Conections}} \quad (2)$$

$$\text{Detection Rate} = 1 - \frac{\text{Number of False Negatives}}{\text{Total Number of Attack Conections}} \quad (3)$$

بنابراین نرخ خطای نادرست مثبت (منفی) عبارت است از درصدی از اتصالات نرمال (حمله) که به‌غلط، برچسب حمله (نرمال) به آن‌ها نسبت داده می‌شود. در آزمایشات انجام شده، از دو معیار نرخ تشخیص و نرخ خطای مثبت نادرست برای ارزیابی راهکار ارائه‌شده استفاده شده است.

۴- اتوماتاهای یادگیر

اتوماتای یادگیر [8] ماشینی است که می‌تواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود، نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تأثیر می‌گیرد. هدف نهایی این است که اتوماتا یاد بگیرد تا از بین اعمال خود، بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل 1 مشاهده می‌شود.



شکل 1: ارتباط بین اتوماتای یادگیر و محیط [8]

محیط را می‌توان توسط سه‌تایی $E \equiv \{\alpha, \beta, c\}$ نشان‌داد که $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه‌ورودی‌ها، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه‌خروجی‌ها، و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمال‌های جریمه می‌باشد. هرگاه β مجموعه‌ای دوعضوی باشد، محیط از نوع P است. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع Q، $\beta(n)$ می‌تواند

[5] False Negative Rate

به‌طور گسسته یک مقدار از مقادیر محدود در فاصله [0.1] را اختیار کند و در محیط از نوع S، $\beta(n)$ متغیر تصادفی در فاصله [0.1] است. c_i احتمال این‌که عمل α_i نتیجه نامطلوب داشته باشد می‌باشد. در محیط ایستا، مقادیر c_i بدون تغییر می‌مانند، حال آن‌که در محیط غیرایستا این مقادیر در طی زمان تغییر می‌کنند.

اتوماتای یادگیر با ساختار ثابت توسط پنج‌تایی $\{\alpha,\beta,F,G,\phi\}$ نشان داده می‌شود که در آن، $\alpha \equiv \{\alpha_1,\alpha_2,...,\alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta \equiv \{\beta_1,\beta_2,...,\beta_r\}$ مجموعه ورودی‌های اتوماتا، $\phi(n) \equiv \{\phi_1,\phi_2,...,\phi_k\}$ مجموعه وضعیت‌های داخلی اتوماتا در لحظه n، $F:\phi\times\beta\to\phi$ تابع تولید وضعیت جدید اتوماتا و $G:\phi\to\alpha$ تابع خروجی می‌باشد که وضعیت کنونی اتوماتا را به خروجی بعدی می‌نگارد.

اتوماتای یادگیر با ساختار متغیر را می‌توان توسط چهارتایی $\{\alpha,\beta,p,T\}$ نشان داد که $\alpha \equiv \{\alpha_1,\alpha_2,...,\alpha_r\}$ مجموعه اعمال اتوماتا، $\beta \equiv \{\beta_1,\beta_2,...,\beta_r\}$ مجموعه ورودی‌های اتوماتا، $p=\{p_1,..,p_r\}$ بردار احتمال انتخاب هریک از عمل‌ها و $p(n+1)=T[\alpha(n),\beta(n),p(n)]$ الگوریتم یادگیری می‌باشد. الگوریتم زیر یک نمونه از الگوریتم‌های یادگیری خطی است. فرض می‌کنیم عمل α_i در مرحله nام انتخاب شود.

- پاسخ مطلوب از محیط

$$\begin{aligned} p_i(n+1) &= p_i(n)+a[1-p_i(n)] \\ p_j(n+1) &= (1-a)\,p_j(n) \quad \forall j \quad j\neq i \end{aligned} \tag{4}$$

- پاسخ نامطلوب از محیط

$$\begin{aligned} p_i(n+1) &= (1-b)\,p_i(n) \\ p_j(n+1) &= (b/r-1)+(1-b)\,p_j(n) \quad \forall j \quad j\neq i \end{aligned} \tag{5}$$

در روابط بالا، a پارامتر پاداش و b پارامتر جریمه می‌باشد. با توجه به مقادیر a و b سه حالت را می‌توان درنظر گرفت: اگر a و b با هم برابر باشند، الگوریتم را L_{RP}، هنگامی‌که b از a خیلی کوچکتر باشد، الگوریتم را $L_{R\varepsilon P}$ و اگر b مساوی صفر باشد آن را L_{RI} می‌نامیم.

۵- مدل پیشنهادی برای تشخیص نفوذ

مدلی که برای تشخیص نفوذ ارائه خواهیم داد، شامل چند سطح از عامل‌های مبتنی بر اتوماتای یادگیر می‌باشد که هر سطح مسئول پردازش بخشی از اطلاعات موجود در رکوردهای مجموعه داده *KDD99* است. در واقع مدل طراحی‌شده، ساختاری سلسله مراتبی متشکل از عامل‌ها دارد. داده‌ها از بالاترین سطح وارد شده و تا پایین‌ترین سطح حرکت می‌کنند. در هر سطح، رکورد مورد نظر با توجه به وظیفه نسبت داده‌شده به همان سطح، پردازش شده و به سطح بعدی هدایت می‌گردد. ساختار ارائه شده در حقیقت داده‌ها را دسته‌بندی کرده و آن‌ها را به یکی از دو کلاس نرمال و حمله منسوب می‌کند. بنابراین در این مدل، مسئله تشخیص نفوذ به‌صورت یک مسئله دسته‌بندی دوکلاسه درنظر گرفته می‌شود .

در این‌جا برای تشخیص نفوذ، از روش تشخیص ناهنجاری استفاده می‌کنیم. به این معنا که دسته‌بندی‌کننده، در مرحله آموزش با استفاده از نمونه‌های متعددی از رفتار نرمال سیستم، الگوی رفتار نرمال را یاد می‌گیرد و در مرحله آزمایش، هرگونه رفتار و فعالیتی در سیستم را که از رفتار نرمال تخطی کند، به‌عنوان حمله تشخیص می‌دهد. بنابراین، به‌عنوان مجموعه آموزش، از زیرمجموعه‌ای از مجموعه داده‌های *10-percent* شامل رکوردهای مربوط به فعالیت نرمال، و به‌عنوان مجموعه آزمایش، از زیرمجموعه‌ای از مجموعه داده‌های *corrected* شامل رکوردهای مربوط به فعالیت نرمال و حملات *DoS* استفاده می‌نماییم .

برای ساده‌سازی، کاهش ابعاد و همچنین افزایش سرعت و کلیت مدل ، به جای استفاده از تمام ویژگی‌های مجموعه داده *KDD99* ، از 12/195٪ ویژگی‌ها (5 ویژگی از مجموعه 41 ویژگی) استفاده می‌کنیم. برای انتخاب این 5 ویژگی، از معیاری که در [9] برای کاهش بعد در مجموعه داده‌های بزرگ معرفی شده استفاده می‌نماییم. این معیار، نوعی بهره اطلاعاتی علامت‌دار است که خود بر اساس دو معیار حساسیت عدم تطابق و دقت عدم تطابق و به کمک *Scree Test* و *Critical Eigenvalue Test* تعریف می‌شود. براساس این معیار، 5 ویژگی از میان 41 ویژگی مجموعه داده‌ها استخراج شده و در دسته‌بندی مورد استفاده قرار می‌گیرند. ویژگی‌هایی که با استفاده از این معیار انتخاب می‌شوند عبارتند از: *src_bytes*، *dst_bytes*، *duration*، *is_host_login* و *is_guest_login*. البته تعداد ویژگی‌هایی که انتخاب می‌شوند قابل تغییر بوده و افزایش یا کاهش آن، بر روی سرعت و دقت دسته‌بندی‌کننده مؤثر است .

علاوه بر کاهش ابعاد، برای افزایش سرعت، کاهش بار محاسباتی و کاهش پیچیدگی دسته‌بندی‌کننده، از مدل ظرف و گوی[6] که در [10] ارائه گردیده استفاده نموده و مقادیر ویژگی‌ها را گسسته می‌کنیم. این مدل دارای دو پارامتر N و M می‌باشد که به‌ترتیب تعداد ظرف‌ها و تعداد گوی‌ها در هر ظرف را نشان می‌دهند. در دسته‌بندی‌کننده پیشنهادی، پارامتر N یا تعداد ظرف‌ها، متناظر با 5 ویژگی انتخاب‌شده برای دسته‌بندی درنظر گرفته می‌شود. همچنین پارامتر M یا تعداد گوی‌ها در هر ظرف را برابر با تعداد مقادیر ویژگی متناظر با آن ظرف پس از گسسته‌سازی درنظر می‌گیریم .

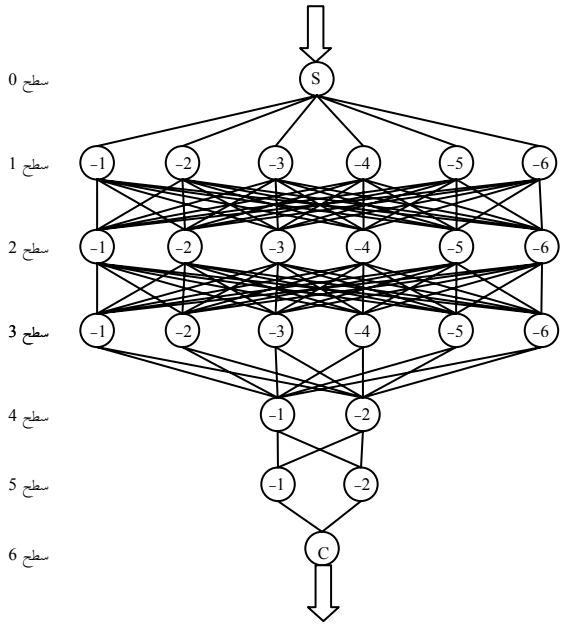
در حالتی که مقادیر ویژگی گسسته باشد، تعیین M برای آن ویژگی آسان است؛ اما هنگامی‌که ویژگی دارای مقادیر پیوسته باشد، M ممکن است بسیار بزرگ شده و دسته‌بندی را غیرممکن سازد. برای حل این مشکل، مقادیر هریک از ویژگی‌های پیوسته را براساس روش ارائه شده در [11] به 6 بازه با اندازه‌های مساوی تقسیم می‌نماییم. تعداد

[6] Urn and Ball Model

این بازه‌ها با موازنه بین سرعت و دقت دسته‌بندی‌کننده قابل تغییر است. بنابراین، ویژگی‌ها را با نماد 1،...،5 و مقادیر آن‌ها را در صورت پیوسته بودن ویژگی با نماد 1،...،6 و درصورت گسسته بودن ویژگی با نماد 1،... (تا تعداد مقادیر مجزای آن ویژگی) نشان می‌دهیم.

از بین 5 ویژگی انتخاب شده، ویژگی اول، دوم و سوم دارای مقادیر پیوسته می‌باشند؛ بنابراین برای این ویژگی‌ها، M را برابر با 6 درنظر گرفته و مقادیر را به 6 بازه گسسته می‌نماییم. ویژگی چهارم و پنجم دودویی بوده و دارای دو مقدار 0 و 1 می‌باشند. به همین دلیل مقدار M برای این ویژگی‌ها برابر با 2 خواهد بود. بنابراین هر اتصال را می‌توان به‌صورت رشته‌ای از نمادها که متناظر با مقادیر ویژگی‌های آن اتصال می‌باشند درنظر گرفت. هر رشته در واقع مشاهده‌ای از وضعیت سیستم است که باید به‌صورت حمله و یا نرمال دسته‌بندی گردد.

در مدل پیشنهادی برای تشخیص نفوذ، درصورتی که پارامترهای مربوط به وضعیت نرمال که در فاز آموزش به مدل آموزش داده شده‌اند، با احتمال بیشتری یک مشاهده از وضعیت سیستم را شناسایی کنند، آن مشاهده توسط دسته‌بندی‌کننده به‌عنوان یک وضعیت نرمال درنظر گرفته می‌شود و بالعکس. در این دسته‌بندی‌کننده، آن‌چه مسئول یادگیری پارامترهای لازم برای محاسبه احتمال تعلق یک رشته به کلاس نرمال یا حمله است، اتوماتای یادگیر می‌باشد. شکل2 شمای کلی دسته‌بندی‌کننده را نشان می‌دهد.



شکل2: مدل پیشنهادی برای تشخیص نفوذ

در مرحله آزمایش، عامل S و عامل‌های سطوح میانی، عملکردی مشابه مرحله آموزش دارند و فقط در مرحله آزمایش دیگر لازم نیست اتوماتای یادگیر این عامل‌ها بهنگام شود. اما عامل C در مرحله آزمایش متفاوت با فاز آموزش عمل می‌کند. در فاز آزمایش، تنها وظیفه عامل C این است که نتیجه دسته‌بندی را با مقایسه نتیجه پردازش رشته با T تعیین نماید.

در ادامه، عملکرد هر عامل را در فاز آموزش و آزمایش به‌طور دقیق به‌صورت شبه‌کد بیان می‌کنیم. در این شبه‌کد، R نشان‌دهنده رکورد یا رشته ورودی می‌باشد. x نمادی از رشته است که در هر لحظه پردازش می‌شود. i نشان‌دهنده سطح و j نشان‌دهنده شماره عامل در هر سطح می‌باشد؛ بنابراین عامل (i,j)، عامل jام از سطح iام خواهد بود. k نشان‌گر مرحله الگوریتم است که در حقیقت نشان می‌دهد رکوردی که درحال پردازش است چندمین رکوردی است که پردازش می‌شود.

تابع $x=Shift(R)$ اولین کاراکتر رشته R از سمت چپ را در متغیر x قرار داده و R را یک کاراکتر به سمت چپ شیفت می‌دهد. تابع $R^{new}=Append(R^{old},x)$ کاراکتر x را به انتهای سمت راست R متصل کرده و رشته به‌دست‌آمده را مجدداً در R قرار می‌دهد. این تابع کمک می‌کند که رشته R در حین پردازش از بین نرفته و در انتها برای تعیین مسیری که برای پردازش رشته طی شده مورد استفاده قرار گیرد.

P_N نشان‌دهنده احتمالی است که توسط مدل برای نرمال بودن یک رشته محاسبه می‌شود و از حاصل‌ضرب احتمال یال‌هایی که برای پردازش رشته توسط مدل طی می‌شود به‌دست می‌آید. بنابراین، برای دسته‌بندی یک رشته باید P_N را با پارامتر T مقایسه نمود.

➢ *الگوریتم مرحله آموزش*

0. تنظیمات اولیه
   - 0-1- $k \leftarrow 1$ ، $P_N \leftarrow 1$ و $T \leftarrow 0$.
   - 0-2- برای هر اتوماتای یادیگر، بردار احتمال به‌صورت زیر مقداردهی اولیه می‌شود:
     - اگر تعداد اعمال اتوماتا برابر با 6 باشد، مقدار اولیه احتمال هر عمل برابر با 1/6 قرار داده می‌شود.
     - اگر تعداد اعمال اتوماتا برابر با 2 باشد، مقدار اولیه احتمال هر عمل برابر با 1/2 قرار داده می‌شود.
1. رکورد kام از مجموعه آموزش از سطح 0 تا سطح 6 دسته‌بندی‌کننده حرکت کرده و در هر سطح مراحل مربوط به آن سطح روی رکورد اجرا می‌شود.
   - سطح 0، عامل S:
     - مقدار 5 ویژگی *src_bytes*، *dst_bytes*، *duration*، *is_host_login* و *is_guest_login* از رکورد ورودی انتخاب شده و در R قرار داده می‌شود.
     - مقادیر ویژگی‌های *src_bytes*، *dst_bytes* و *duration* (ویژگی‌های 1، 2 و 3) به 6 بازه

گسسته شده و در R با نماد 1 تا 6 جایگزین می‌شوند.

- مقادیر ویژگی‌های *is_host_login* و *is_guest_login* (ویژگی‌های 4 و 5) در R با نماد 1 (معادل مقدار 0) یا 2 (معادل مقدار 1) جایگزین می‌شوند.
- *x=Shift(R)*
- $R^{new} = Append(R^{old}, x)$.
- اگر مقدار x برابر با j باشد، $P_N^{new} \leftarrow P_N^{new} \times p(j)$ و R به عامل jام از سطح 1 ارسال می‌شود.

- سطح i $(1 \le i \le 4)$، عامل *(i-j)*:
  - *x=Shift(R)*
  - $R^{new} = Append(R^{old}, x)$.
  - اگر مقدار x برابر با j باشد، $P_N^{new} \leftarrow P_N^{new} \times p(j)$ و R به عامل jام از سطح $i+1$ ارسال می‌شود.
- سطح 5، عامل *(5-j)*:
  - R و P_N به عامل C ارسال می‌شوند.
- سطح 6، عامل *C*:
  - اگر $T=0$، $T \leftarrow P_N$. در غیر این صورت

$$T^{new} \leftarrow T^{old} + \frac{1}{k}(P_N - T^{old}) \qquad (6)$$

  - عامل C سیگنال‌های پاداش را به عامل‌های دارای اتوماتاهای یادگیر موجود در مسیری که توسط رشته R مشخص می‌شوند ارسال می‌کند. برای این کار:
    i. به عامل S سیگنالی حاوی $R(1)$ و مقدار پاداش ارسال می‌گردد. این سیگنال به این معناست که عامل S می‌بایست احتمال عمل $R(1)$ام از اتوماتای یادگیر خود را افزایش دهد.
    ii. به ازای هر i $(1 \le i \le 5)$، سیگنالی حاوی $R(i+1)$ و مقدار پاداش به عامل $(i - R(i))$ ارسال می‌گردد. این سیگنال به این معناست که عامل $(i - R(i))$ می‌بایست احتمال عمل $R(i+1)$ از اتوماتای یادگیر خود را افزایش دهد.
  - $k \leftarrow k+1$.

2. درصورتی‌که شرط خاتمه برقرار نباشد، به مرحله 1 بازمی‌گردیم.
3. پایان الگوریتم.

➢ *الگوریتم مرحله آزمایش*

0. تنظیمات اولیه
   - $k \leftarrow 1$ ، $P_N \leftarrow 1$ و $T \leftarrow 0$.
1. رکورد kام از مجموعه آموزش از سطح 0 تا سطح 6 دسته‌بندی‌کننده حرکت کرده و در هر سطح مراحل مربوط به آن سطح روی رکورد اجرا می‌شود.
   - سطح 0، عامل *S*:
     - مقدار 5 ویژگی *src_bytes*، *dst_bytes*، *duration*، *is_host_login* و *is_guest_login* از رکورد ورودی *(R)* انتخاب می‌شود.
     - مقادیر ویژگی‌های *src_bytes*، *dst_bytes* و *duration* (ویژگی‌های 1، 2 و 3) به 6 بازه گسسته شده و در R با نماد 1 تا 6 جایگزین می‌شوند.
     - مقادیر ویژگی‌های *is_host_login* و *is_guest_login* (ویژگی‌های 4 و 5) در R با نماد 1 (معادل مقدار 0) و 2 (معادل مقدار 1) جایگزین می‌شوند.
     - *x=Shift(R)*
     - $R^{new} = Append(R^{old}, x)$.
     - اگر مقدار x برابر با j باشد، $P_N^{new} \leftarrow P_N^{new} \times p(j)$ و R به عامل jام از سطح 1 ارسال می‌شود.
   - سطح i $(1 \le i \le 5)$، عامل *(i-j)*:
     - *x=Shift(R)*
     - $R^{new} = Append(R^{old}, x)$.
     - اگر مقدار x برابر با j باشد، $P_N^{new} \leftarrow P_N^{new} \times p(j)$ و R به عامل jام از سطح $i+1$ ارسال می‌شود.
   - سطح 5، عامل *(5-j)*:
     - R و P_N به عامل C ارسال می‌شوند.
   - سطح 6، عامل *C*:
     - اگر $P_N \ge T$، رکورد ورودی متعلق به کلاس نرمال و در غیر این صورت مربوط به کلاس حمله است.
     - $k \leftarrow k+1$.
2. درصورتی‌که شرط خاتمه برقرار نباشد، به مرحله 1 بازمی‌گردیم.
3. پایان الگوریتم.

در مرحله آموزش، پس از طی مسیر مربوط به هر رشته (نرمال) روی دسته‌بندی‌کننده، پاداشی به اتوماتاهای یادگیری که در آن مسیر

دخالت داشته‌اند تعلق می‌گیرد تا احتمال آن مسیر افزایش یابد. همان‌طور که گفته شد، متغیر P_N نشان‌دهنده احتمال مسیر است و به‌صورت حاصل‌ضرب بخش‌های مختلف مسیر (یال‌های موجود در مسیر) محاسبه می‌شود. احتمال هر بخش یا هر یال از مسیر به این معناست که در رشته ورودی تا چه حد محتمل است که پس از نماد ابتدایی آن یال، نماد انتهایی آن یال قرار گیرد. بنابراین افزایش احتمال هر مسیر نشان می‌دهد که رشته تولیدشده توسط آن مسیر، با احتمال بالاتری نسبت به قبل نشان‌دهنده یک فعالیت نرمال است؛ زیرا یک نمونه از چنین رشته‌ای، در مجموعه آموزش که شامل نمونه رشته‌های مربوط به فعالیت نرمال می‌باشد مشاهده شده‌است.

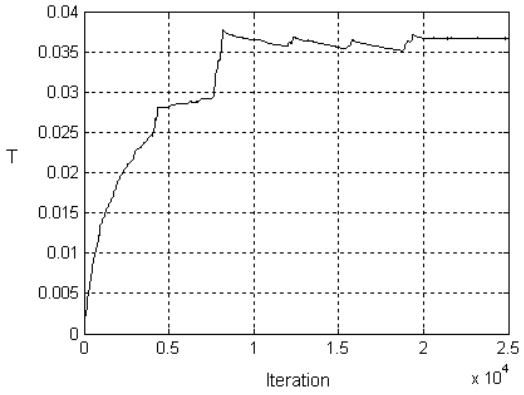
تنظیم حد آستانه در مرحله آموزش به این صورت است: در صورتی‌که $P_N > T$ باشد، مدل قادر به تشخیص صحیح رشته بوده و آن را به‌صورت یک فعالیت نرمال دسته‌بندی کرده است. اما برای این‌که از کوچک بودن بیش از حد T جلوگیری کنیم، با استفاده از رابطه 3-4، مقدار T را افزایش می‌دهیم تا مدل، رشته‌هایی با احتمال بیش از حد کوچک اما بزرگتر از T را به‌عنوان نرمال شناسایی نکند.

در حالتی که $P_N < T$ باشد، در حقیقت دسته‌بندی‌کننده نتوانسته تشخیص صحیحی برای رشته ارائه دهد و آن را تحت عنوان حمله شناسایی کرده است. بنابراین T باید کاهش یابد تا مدل، این رشته ورودی را نیز شامل شده و آن را جزء حالات نرمال سیستم درنظر بگیرد. این کار با استفاده از رابطه زیر صورت می‌گیرد.

$$T^{new} \leftarrow T^{old} - \frac{1}{k}(T^{old} - P_N) \qquad (7)$$

همان‌طور که مشاهده می‌شود، رابطه 3-5 دقیقاً معادل رابطه 3-4 می‌باشد و به همین علت در هر دو حالت از رابطه 3-4 برای به‌روزرسانی T استفاده می‌کنیم. حالات دیگری که ممکن است روی دهد این است که $P_N = T$ باشد. در چنین وضعیتی، نیازی به تغییر T وجود ندارد؛ زیرا مدل، دقیقاً رشته‌ای با احتمال P_N را شامل می‌شود و مجدداً می‌توان از رابطه ارائه شده برای به‌روزرسانی T استفاده نمود. در این حالت، $T^{new} \leftarrow T^{old}$ قرار داده می‌شود. از آن‌جا که ضریب $\frac{1}{k}$، با افزایش تعداد رشته‌های ورودی کوچک و کوچک‌تر شده و به‌تدریج به سمت صفر میل می‌کند، T در نهایت به مقدار ثابتی همگرا می‌شود و تغییرات T هنگامی که k بسیار بزرگ شود، تقریباً به صفر می‌رسد.

برای ارزیابی مدل، ابتدا مقادیر اولیه پارامترهای اتوماتای یادگیر، یعنی نرخ یادگیری و سیگنال پاداش راتنظیم کرده و الگوریتم مرحله آموزش را اجرا می‌کنیم. به‌ازاء مقدار 0/01 برای نرخ یادگیری و 0/1 برای پاداش، تغییرات T در مرحله آموزش در شکل3 مشاهده می‌شود. مطابق شکل، مقدار نهایی T برابر با 0/0367 می‌باشد.



شکل3: تغییرات حد آستانه (T) در طول مرحله آموزش مدل

در مرحله آزمایش، هر رشته با توجه به دنباله نمادهای آن، روی مدل حرکت کرده و احتمال مسیری که رشته را تولید می‌کند (P_N) محاسبه می‌گردد. درصورتی‌که این احتمال از حد آستانه‌ای که نرمال بودن یک رشته را نشان می‌دهد بیشتر باشد، رشته به‌عنوان یک فعالیت نرمال تشخیص داده می‌شود. بالعکس اگر احتمال به‌دست‌آمده از حد آستانه کوچک‌تر باشد، آن رشته به‌عنوان یک حمله *DoS* شناسایی می‌گردد.

برای بررسی کارایی روش ارائه شده، مقدار خطای مثبت نادرست و نرخ تشخیص روش پیشنهادی را با تعدادی از روش‌ها که در مراجع مختلف معرفی شده‌اند مقایسه نمودیم. این روش‌ها همگی از مجموعه داده‌های *KDD99* برای آموزش و آزمایش دسته‌بندی‌کننده استفاده می‌نمایند. در [11] روشی مبتنی بر مدل مخفی مارکف ارائه شده که از روشی مشابه مدل پیشنهادی برای انتخاب ویژگی‌ها و گسسته‌سازی آن‌ها استفاده می‌کند. بنابراین به‌دلیل مشابهت شرایط، به‌خوبی می‌توان عملکرد آن را با مدل پیشنهادی مقایسه نمود.

به‌علاوه، مدل پیشنهادی با مدلی مبتنی بر شبکه عصبی پرسپترون چندلایه که در [12] ارائه شده مقایسه شده‌است. برای آن‌که بتوان عملکرد این مدل را با مدل پیشنهادی مقایسه کرد، شبکه عصبی مجدداً پیاده‌سازی شده و برای آموزش و آزمایش آن، مجموعه‌های آموزش و آزمایش را مطابق با آن‌چه در روش پیشنهادی استفاده شده است، تولید می‌کنیم. مجموعه داده‌های وردوی شبکه، زیرمجموعه‌ای از مجموعه داده *10-percent* است که تنها شامل رکوردهای مربوط به حمله *DoS* و نرمال (به‌ترتیب کلاس 0 و 1) از مجموعه *10-percent* می‌باشد. سپس 5 ویژگی مشابه مدل پیشنهادی، از این زیرمجموعه انتخاب و گسسته‌سازی می‌شوند. داده‌های آزمایش نیز شامل رکوردهای *DoS* و نرمال از مجموعه *corrected* می‌باشند که 5 ویژگی از آن‌ها انتخاب شده و گسسته می‌گردند. لازم به ذکر است که شبکه عصبی مورد استفاده، دارای 2 لایه مخفی است. نتایج آزمایشات نشان داد که شبکه عصبی حدوداً 4 برابر کندتر از مدل پیشنهادی عمل می‌کند و انتخاب ویژگی و گسسته‌سازی، موجب ایجاد خطای زیادی در عملکرد دسته‌بندی‌کننده می‌شود.

روش دیگری که برای ارزیابی روش پیشنهادی مورد استفاده قرار گرفته‌است، مدل ارائه شده در [13] می‌باشد که براساس درخت تصمیم، داده‌ها را دسته‌بندی می‌کند. برای مقایسه درخت تصمیم با روش پیشنهاد شده، باید آن‌ها را در شرایط مشابهی پیاده‌سازی کرده و مورد ارزیابی قرار داد. بدین منظور، از مجموعه‌های آموزش و آزمایشی مشابه با آن‌چه در مورد شبکه عصبی توضیح دادیم، استفاده شده است.

نتیجه این مقایسه‌ها در جدول2 مشاهده می‌شود. دلیل انتخاب درخت تصمیم و شبکه عصبی برای مقایسه و ارزیابی عملکرد مدل ارائه شده، کارایی بالای این روش‌ها در تشخیص نفوذ می‌باشد که در مراجع مختلف به آن اشاره شده است. به‌عنوان مثال، برندگان اول تا سوم مسابقه *KDD Cup 1999*، همگی برای تشخیص نفوذ از درخت تصمیم استفاده نموده‌اند. انتخاب ویژگی‌های کم و گسسته‌کردن مقادیر آن‌ها، نرخ تشخیص سیستم را کاهش داده و خطای دسته‌بندی را افزایش می‌دهد؛ اما در مقابل، سرعت سیستم تشخیص نفوذ افزایش قابل توجهی می‌یابد و این مسئله در کاربردهای بلادرنگ بسیار حائز اهمیت است.

جدول 2- مقایسه کارایی مدل پیشنهادی با چند روش تشخیص نفوذ دیگر

روش	نرخ خطای مثبت نادرست	نرخ تشخیص
مدل پیشنهادی	٪12/39	٪87
مدل مخفی مارکف [11]	٪21	٪79
شبکه عصبی پرسپترون چندلایه [12]	٪15/68	٪83/1
درخت تصمیم [13]	٪19/73	٪80/06

۶- نتیجه گیری

در این مقاله، یک سیستم چند عامله مبتنی بر اتوماتای یادگیر برای تشخیص نفوذهای از نوع انکار سرویس که یکی از حملات مهم و شایع در سیستم‌های کامپیوتری است پیشنهاد گردید. عاملها در سیستم چندعامله پیشنهادی از اتوماتاهای یادگیر برای ایجاد هماهنگی بین خود استفاده میکنند. در مدل پیشنهادی ، هر عامل دارای یک اتوماتای یادگیر است که پارامترهای مدل را نگهداری کرده و آن‌ها را با توجه به بازخوردهایی که از محیط دریافت می‌کنند به‌روز می‌رسانند. در واقع در مدل ارائه شده، مسئله تشخیص نفوذ به‌صورت یک مسئله دسته‌بندی دوکلاسه مطرح گردیده است. برای آموزش و آزمایش این مدل، از مجموعه داده‌های *KDD99* استفاده گردید

به‌منظور ساده‌سازی و افزایش کلیت سیستم تشخیص نفوذ، از بین 41 ویژگی موجود در داده‌ها، 5 ویژگی که بیشترین تأثیر را در متمایز کردن نمونه‌ها دارند استخراج شده و همچنین، مقادیر ویژگی‌ها به تعدادی بازه گسسته می‌شوند. مدل پیشنهادی برای تشخیص نفوذ، ابتدا در مرحله آموزش، داده‌های ورودی را دریافت کرده و پس از انجام پیش‌پردازش، آن‌ها را آموزش می‌بینند. سپس در مرحله آزمایش، داده‌های ورودی جدید را با توجه به آن‌چه یاد گرفته دسته‌بندی کرده و به یکی از دو کلاس نرمال یا حمله نسبت می‌دهد. همان‌طور که انتظار می‌رود، پیش‌پردازش (کاهش ویژگی‌ها و گسسته‌سازی آن‌ها)، نرخ تشخیص سیستم را تاحدی کاهش می‌دهد؛ اما در مقابل، سرعت، کلیت و سادگی آن افزایش چشم‌گیری می‌یابد. در مقایسه با چند روش رایج در تشخیص نفوذ، انجام پیش پردازش، کاهش کمتری در نرخ تشخیص سیستم پیشنهادی ایجاد می‌کند.

ازجمله مزایای مدل ارائه‌شده، سادگی محاسبات، کلیت، سرعت بالا و قابلیت استفاده از آن به‌صورت برخط می‌باشد. به‌علاوه، از آن‌جا که پارامترهای سیستم به‌طور خودکار در طول فاز آموزش تنظیم می‌گردند، نیازی به تنظیم دستی آن‌ها نبوده و دقت سیستم و قابلیت تطبیق آن با شرایط جدید افزایش می‌یابد.

سپا سگزاری: این کار تحقیقاتی توسط مرکز تحقیقات مخابرات ایران حمایت مالی شده است که از این طریق سپاسگزاری می گردد.

مراجع

[1] *R. P. Lippman, D. J. Fried, I. Graf and J. W. Haines, "Evaluating Intrusion Detection Systems: the 1998 DARPA Off-line Intrusion Detection Evaluation", Proceedings of DARPA Information Survivability Conference and Exposition, vol. 2, pp. 12 – 26, 2000.*

[2] *S. J. Stolfo et. al., KDD99 Data Set, Available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html*

[3] *H. G. Kayacik, A. N. Zeineir-Heywood and M. I Heywood, "On Dataset Biases in Learning System with Minimum Apriori Information for Intrusion Detection", Proceeding of the 2nd Annual Conference on Communication Networks and Services Research, 2004.*

[4] *W. Lee, "A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems", ACM Transactions of Information and System Security, vol. 3, pp. 227-261, 2000.*

[5] *W. Lee and S. J. Stoflo, "Data Mining Approaches for Intrusion Detection", Proceedings of the 7th USENIX Security Symposium, pp. 79-93, San Antonio, 1998.*

[6] *H. G. Kayacik, A. N. Zeineir-Heywood and M. I Heywood, "On Dataset Biases in Learning System with Minimum Apriopi Information for Intrusion Detection", Proceeding of the 2nd Annual Conference on Communication Networks and Services Research, 2004.*

[7] *J. Allen, "State of the Practice of Intrusion Detection Technologies", Technical Report, CMU, 2000.*

[8] *K. S. Narendra, M. A. L. Thathachar، "Learning automata: An introduction"; Prentice Hall, 1989.*

[9] *G. Kuchimanchi, V. V. Phoha, K. S. Balagani and S. R. Gaddam, "Dimension Reduction using Feature Extraction Methods for Real-time Misuse Detection Sustems", Proceedings of the IEEE Workshop on Information Assurance and Security, 2004.*

[10] *L. A. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" Proccedings of the IEEE, vol.77, pp. 257-286, 1989.*

[11] *S. S. Joshi and V. V. Phoha, "Investigating Hidden Markov Models Capabilities in Anomaly Detection",*

Proceedings of the 43rd Annual Southeast Regional Conference, vol. 1, pp. 98-103, 2005.

[12] *Y. Yao, Y. Wei, F. X. Gao and G. Yu, "Anomaly Intrusion Detection Approach Using Hybrid MLP/CNN Neural Network", Sixth International Conference on Intelligent Systems Design and Applications, vol. 2, pp. 1095 – 1102, 2006.*

[13] *J. H. Lee, J. H. Lee, S. G. Sohn, J. H. Ryu and T. M. Chung, "Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System", 10th International Conference on Advanced Communication Technology (ICACT 2008), 2008.*