فصلنامه علمی - پژوهشی فرماندهی و کنترل، سال پنجم، شماره دو، تابستان 1400

بهره گیری از طبقه‌بندی‌کننده‌ی عصبی– فازی برای سیستم های تشخیص نفوذ

محمد حسن نتاج صلحدار[1]

تاریخ دریافت: ۱۳۹۹/۱۲/۰۸ **تاریخ پذیرش:** ۱۴۰۰//۰۶/۲۲

چکیده

یکی از ابزارهای هوش مصنوعی، سیستم استنتاج تطبیق‌پذیر عصبی – فازی است، که در این مقاله برای ساخت سیستم تشخیص نفوذ مورد استفاده قرار گرفته است و ما آن را دسته‌بندی کننده‌ی عصبی – فازی می‌نامیم. سیستم تشخیص نفوذ مبتنی بر طبقه‌بندی کننده شبکه عصبی-فازی یک سیستم تشخیص نفوذ مبتنی بر ناهنجاری است که از منطق فازی و شبکه عصبی برای تشخیص اینکه فعالیت مخرب در یک شبکه انجام می شود، استفاده می کند. در این مقاله به تشریح معماری دسته‌بندی کننده‌ی عصبی – فازی و مؤلفه های آن می پردازیم. قوانین فازی نمونه برای برخی از حملات ایجاد شده و نتایج آزمایش با داده های واقعی شبکه شرح داده شده است. آزمایشات و ارزیابی های ما با مجموعه NSLKDD مجموعه داده های تشخیص نفوذ انجام شده است که نسخه ای از مجموعه داده های ارزیابی نفوذ KDD Cup99 است که توسط آزمایشگاه های MIT Lincoln تهیه و مدیریت شده است. در نهایت، این مقاله سعی بر آن دارد تا با بررسی عملکرد مدل "سیستم استنتاج تطبیق‌پذیر عصبی – فازی" بر روی یک مجموعه استاندارد و جامع میزان کارایی مدل طراحی شده را نشان دهد.

واژگان کلیدی: سیستم تشخیص نفوذ، شبکه عصبی، طبقه‌بندی کننده‌ی عصبی – فازی، انفیس، NSLKDD

[1] دانشگاه شهید چمران اهواز_پردیس صنعتی شهدای هویزه، اهواز، ایران. مربی. n.solhdar@scu.ac.ir
نویسنده مسئول : محمد حسن نتاج صلحدار

مقدمه

سیستم‌های تشخیص نفوذ[1](IDS) در حال حاضر جزء اصلی‌ترین و کامل‌ترین قسمت‌های یک سیستم پایش شبکه می‌باشند. سیستم‌های تشخیص نفوذ فناوری‌های تقریبا جدیدی هستند و این نوید را به ما می‌دهند که به ما در جهت شناسایی نفوذهایی که به شبکه انجام می‌شود کمک خواهند کرد[1]. تشخیص نفوذ در واقع فرآیندی است که در آن رویدادها و رخدادهای یک سیستم یا شبکه پایش شده و براساس این پایش‌ها وقوع نفوذ به آن شبکه یا سیستم تشخیص داده می‌شود. یک نفوذ در واقع فعالیت یا عملی است که توسط آن محرمانگی، صحت و تمامیت و یا دسترس پذیری به منابع دچار اختلال و یا تعرض می‌شود[2].

IDSها نیاز به جمع آوری داده های کافی برای ساختن یک مدل ریاضی پیچیده دارند، که در مورد ترافیک شبکه پیچیده غیر عملی است [3]. و همچنین تشخیص نفوذهای جدید برای IDSها کار دشواری است، و پایگاه داده مبتنی بر امضا، باید به صورت دستی و مکرر به روز شود [4]. برای حل محدودیت های روش های فوق، تعدادی از تکنیک های داده کاوی معرفی شده اند [5]. در بین این تکنیک ها، شبکه عصبی یکی از تکنیک‌های است که بسیار مورد استفاده قرار میگیرد و با موفقیت در تشخیص نفوذ به کار گرفته شده است[6]. با توجه به انواع مختلف شبکه عصبی، این تکنیک ها را می‌توان در سه دسته زیر طبقه بندی کرد: تشخیص نفوذ مبتنی بر شبکه عصبی نظارت شده، تشخیص نفوذ مبتنی بر شبکه عصبی بدون نظارت و تشخیص نفوذی مبتنی بر شبکه عصبی ترکیبی.

در این پژوهش، از سیستم استنتاج تطبیق پذیر عصبی – فازی، برای ساخت سیستم تشخیص نفوذ استفاده شده است. این سیستم برای بررسی مسائلی با معادلات غیرخطی بسیار کاربردی است[7]. دسته‌بندی‌کننده‌ی عصبی – فازی، ترکیبی از شبکه‌ی عصبی و سیستم استنتاج فازی است. یکی از مشکلات سیستم استنتاج فازی، عدم قابلیت یادگیری است. از طرف دیگر، شبکه‌ی عصبی قابلیت یادگیری دارد و می‌تواند بارها و بارها آموزش دیده و خود را مطابق آموزش‌های جدید برنامه-ریزی کند. با ترکیب شبکه‌ی عصبی با قابلیت یادگیری و سیستم استنتاج فازی، به یک شبکه‌ی یادگیرنده می‌رسیم که هم قابلیت یادگیری دارد و هم قابلیت کارکردن با قوانینی دارد که مطابق قوانین سیستم استنتاج فازی هستند. همچنین در این تحقیق سعی شده است از شبکه عصبی – فازی به عنوان یک طبقه‌بندی‌کننده استفاده شود. در این مدل ابتدا قوانین فازی لازم بدون نیاز به یک فرد خبره و با استفاده از روش خوشه‌بندی کاهشی بر اساس مجموعه‌داده‌های آموزشی[2]انتخاب شده از مجموعه‌داده-های[3] NSLKDD تولید می‌شود، سپس قوانین فازی بدست‌آمده برای ایجاد یک شبکه عصبی – فازی تطبیق پذیر به نام انفیس[4]به کارگرفته می‌شوند. اندازه‌ی قدرت هر قانون فازی در طی فرآیند آموزش شبکه‌ی عصبی – فازی محاسبه و در طول پروسه‌ی آموزش، با تنظیم مجدد توابع عضویت[5]که سازنده‌ی قوانین فازی هستند، شبکه به سمت خطای کمتر گام برمی‌دارد.

در نهایت مدل بدست آمده به عنوان طبقه‌بندی‌کننده، برای کلاس‌بندی داده‌های آزمایشی[6] که همانند داده‌های آموزشی از مجموعه‌داده‌های NSLKDD استخراج شده-اند به کارگرفته می‌شود. عدم کفایت یا مناسب بودن این

1 Intrusion Detection Systems (IDS)
2 Learning data
3 International Conference on Knowledge Discovery and Data Mining
4 Adaptive Network-Based Fuzzy Inference System (ANFIS)
5 Membership Function .
6 Testing data

مدل برای انجام عملیات طبقه‌بندی مورد بررسی قرار می-گیرد. طبقه‌بندی‌کننده‌ی ارائه شده، به صورت یک طبقه-بندی‌کننده‌ی دوگانه که رکوردهای اتصال را به دو کلاس نفوذ و عادی تقسیم می‌کند، استفاده خواهدشد و داده‌های آموزش و آزمایش مورد طبقه‌بندی قرار خواهندگرفت. در ادامه این مقاله کارهای که در این حوزه انجام شده، آورده شده است و در روش پیشنهادی الگوریتم دسته بندی کننده فازی، توضیح داده شده است و در ادامه پیاده سازی و نتایج آماری آورده شده است و بعد از آن در مورد معیارهای ارزیابی که در این تحقیق از آن استفاده شد توضیحاتی داده شده است و در انتها مقایسه و نتیجه گیری آمده است.

کارهای انجام شده

اخیراً، روش های محاسبات نرم برای سیستم های تشخیص نفوذ استفاده می شود. برخی از این روشها در مجموعه داده KDD مورد ارزیابی قرار گرفته است. طبقه بندی کننده های مبتنی بر قانون فازی[1]، درخت تصمیم گیری[2]، ماشین های بردار پشتیبانی[3]، برنامه نویسی ژنتیکی[4]خطی در [8] مورد استفاده قرار گرفته‌شده‌اند تا اهمیت نمونه های محاسبات نرم[5]را برای مدل سازی سیستم های تشخیص نفوذ نشان دهند.

IDS مبتنی بر آماری، از روشهای آماری متنوعی از جمله تجزیه و تحلیل مؤلفه های اصلی[6]، تجزیه و تحلیل خوشه ای و چند متغیره، آنالیز بیزی استفاده می کند. فارون و بوکلیف[7]and در تحقیق خود با استفاده از خوشه بندی K-means سعی در بهبود آموزش شبکه عصبی کردن به نرخ تشخیص خطا 92 درصدی و نرخ هشدار غلطی 6,21 درصدی دست یافته‌اند[9].

دش[8]در مقاله خود [10] نشان داد که نتایج آزمون شبکه عصبی مصنوعی پیشنهادی قادر به تشخیص ۹۴,۹% حملات هستند. جونالاگادا و رد[9] در تحقیقات خود با به کارگیری دسته‌بندی‌کننده‌های عصبی – فازی در یک معماری دو لایه موفق شدند به نرخ تشخیص خطا 95,3% و نرخ هشدار غلطی 1,9% دست یابند[11]. در [12] از روش TLMD[10] برای سیستم تشخیص نفوذ استفاده شده است و سعی کرد داده های پارازیت دار[11]را کاهش دهد. امبوسایدی[12]و همکارانش [13] نیز از روشهای ماشین بردار پشتیبان خاص با نام LSSTM[13] برای تشخیص نفوذ استفاده کردند. بقداد[14]معتقد است، عملکرد شبکه عصبی بدون نظارت پایین است. به خصوص برای حملات با تعداد کم، شبکه عصبی بدون نظارت همچنین از دقت کمتری برخوردار است. ایشان با به کارگیری شبکه‌های عصبی و ساخت شبکه با پیکربندی‌های مختلف و آموزش-های متعدد شبکه به نرخ تشخیص خطا 84,25 درصد و نرخ هشدار غلطی 15,75 درصد دست یافت[14].

شبکه عصبی ترکیبی، که ترکیبی از شبکه عصبی تحت نظارت[15]و شبکه عصبی بدون نظارت[16]و یا ترکیب شبکه عصبی با سایر تکنیک های داده کاوی برای تشخیص نفوذ است، یکی دیگر از نمونه های محاسبات نرم در سیستم های تشخیص نفوذ است [15]. انگیزه استفاده از شبکه عصبی ترکیبی غلبه بر محدودیتهای شبکهای عصبی تنها است. جیراپومین[17]و همکاران [16] استفاده از شبکه عصبی ترکیبی را هم برای رویت هرگونه حمله با استفاده از SOM کوهنن[18]و هم برای طبقه بندی حمله ها با استفاده از شبکه های عصبی انتشار انعطاف پذیر پیشنهاد کردند. هوریس[19]

1 Fuzzy rule
2 Decision tree
3 Support vector machine
4 genetic programing
5 Soft computing
6 Principal Component Analysis - PCA
7 Faraoun and Boukelif
8 Dash
9 Jonnalagadda and Redd
10 Two Layers Multi-class Detection
11 noise
12 Ambusaidi
13 Least Square Support Vector Machine
14 Beghdad
15 supervised
16 unsupervised
17 Jirapummin
18 Kohenen
19 Horeis

از ترکیبی SOM[1] و شبکه های تابعی (RBF[2]) استفاده کرد. این سیستم به طور کلی نتایج بهتری را از IDS مبتنی بر شبکه های RBF ارائه می دهد [17]. هان و چو[3] برای تعیین ساختار و وزن توالی های تماس، یک روش تشخیص نفوذ را بر اساس شبکه های عصبی تکاملی پیشنهاد کردند [18]. چن[4] و همکارانش IDS مبتنی بر درخت عصبی انعطاف پذیر ترکیبی را بر اساس درخت عصبی انعطاف پذیر، الگوریتم تکاملی و بهینه سازی ذرات (PSO[5]) پیشنهاد دادند. نتایج تجربی نشان داد که روش پیشنهادی کارآمد است [19]. طوسی و کاهانی[6] [20] و قوش[7] و همکارانش [21] در تحقیقات خود برای سیستم تشخیص نفوذ پیشنهادی با استفاده از شبکه فازی به نرخ های تشخیص بهتر از 95,3% و 91,1% دست نیافتند.

در ادامه روش پیشنهادی بیان می شود و در قسمت ارزیابی نتایج با چند روش دیگر مقایسه خواهد گردید.

روش پیشنهادی

شبکه‌ی عصبی مصنوعی و منطق فازی، هر دو ابزارهای هوش مصنوعی هستند که می‌توانند برای ساخت یک سیستم هوشمند دیگر، مکمل یکدیگر باشند. شبکه‌ی عصبی مصنوعی ساختار محاسباتی سطح پایینی است که به خوبی با داده‌های خام اولیه کار می‌کند. در مقابل منطق فازی با استدلال‌های سطح بالایی که با استفاده از دانش یک متخصص در یک حوزه‌ی خاص بدست‌آمده‌اند، سروکار دارد.

یکی از معایب سیستم فازی، عدم توانایی یادگیری است که سبب می‌شود این سیستم‌ها قادر به تطبیق و تنظیم خود با محیط جدید نباشند. از طرف دیگر، اگرچه شبکه‌ی عصبی مصنوعی توانایی یادگیری دارد، اما برای انسان غیرشفاف و مانند یک جعبه سیاه[8] می‌باشد.

ادغام شبکه‌ی عصبی مصنوعی با سیستم فازی، نوید یک سیستم کامل‌تر نسبت به هر دو آن‌ها را می‌دهد. سیستم ترکیبی عصبی – فازی، می‌تواند توانایی‌های شبکه‌ی عصبی از قبیل پردازش موازی و قابلیت یادگیری را با ویژگی‌های سیستم فازی مانند توضیح پذیری و انسان‌پسندانه تر بودن، ادغام نماید. در نتیجه، شبکه‌ی عصبی شفاف‌تر می‌شود و سیستم فازی، قابلیت یادگیری پیدا می‌کند[22].

یک از انواع شبکه‌ی عصبی – فازی، از ترکیب شبکه‌ی عصبی مصنوعی با فازی سوجینو[9] شکل می‌گیرد. در سال 1993 ژانگ[10] برای اولین بار با مدنظر قرار دادن توانایی-های تئوری فازی که مبتنی بر قواعد منطقی بوده و همچنین روش شبکه عصبی مصنوعی که توانائی استخراج دانش از اطلاعات عددی را دارند، سیستم استنتاج تطبیقی عصبی - فازی را ارائه نمود[23]. سیستم ارائه‌شده توسط ژانگ، انفیس خوانده می‌شود. اصطلاح انفیس مخفف عبارت، Adaptive Nero Fuzzy Inference System است که به اختصار، روش استنتاج فازی عصبی تطبیقی نیز نامیده می-شود. در این تحقیق ما از انفیس برای ساخت سیستم تشخیص نفوذ استفاده و از عنوان دسته‌بندی کننده‌ی عصبی – فازی برای آن استفاده خواهیم کرد. دسته‌بندی کننده‌ی عصبی - فازی، به طور گسترده برای بررسی پدیده‌هایی با معادلات غیرخطی به کار گرفته شده است. بنابراین، ترکیب سیستم های فازی که بر قواعد منطقی استوار هستند و روش شبکه‌های عصبی مصنوعی که توان استخراج دانش از اطلاعات عددی را دارند، ما را قادر می سازد تا بتوانیم در کنار استفاده از دانش بشری از اطلاعات موجود نیز، در ساخت مدل استفاده کنیم. یکی از مزایای استفاده از سیستم انفیس این است که پس آموزش شبکه با استفاده نمونه‌های

1 self-organization map
2 Radial base function
3 Han and Cho
4 Chen
5 Particle Swarm Optimization
6 Toosi and Kahani
7 Ghosh
8 Black Box
9 Sugeno
10 Zhang

آموزشی، سیستم نتایج حاصله را در قالب قوانین فازی که به صورت محاوره‌ای، قابل فهم و تحلیل می باشند، ارائه می‌دهد. این ویژگی می‌تواند در ارزیابی قوانین و مقایسه آنها با مفاهیم و قوانین افراد خبره حائز اهمیت باشد.

در سیستم فازی سوجینو قوانین به صورت زیر استنتاج می شوند:

IF x_1 is A_1
AND x_2 is A_2
.....
AND x_m is A_m

THEN $y = f(x_1, x_2, \ldots, x_m)$

لایه‌های دسته‌بندی کننده‌ی عصبی - فازی

در این بخش به تشریح لایه‌های انفیس خواهیم پرداخت. ورودی و خروجی هر لایه مشخص شده و رابطه‌ی مرتبط با آن نیز بیان می‌شود[22].

لایه‌ی 1: لایه‌ی 1 لایه‌ی ورودی است. گره‌ها در این لایه داده‌ها را برای لایه‌ی 2 آماده می‌کنند. در این لایه هیچ تغییری روی داده‌ها صورت نمی‌گیرد بطوری که ورودی با خروجی برابر است.

$$y_i^{(1)} = x_i^{(1)} \qquad (1)$$

لایه‌ی 2: در این لایه عمل فازی سازی روی داده‌ها انجام می شود. در مدل ژانگ، تابع فعالیت برای عمل فازی سازی داده‌ها، تابع زنگوله[1]ست که معادله‌ی آن به صورت زیر است:

$$y_i^{(2)} = \frac{1}{1 + \left(\frac{x_i^{(2)} - a_i}{c_i}\right)^{2b_i}} \qquad (2)$$

در این معادله، $\mathbf{x_i^{(2)}}$ ورودی و $\mathbf{y_i^{(2)}}$ خروجی گره **i**ام در لایه دوم است و پارامترهای **a** و **b** و **c** به ترتیب شیب، عرض و مرکز تابع فعالیت می‌باشند.

لایه‌ی 3: این لایه، لایه‌ی قوانین است. هر گره در این لایه، یک قانون فازی را نشان می‌دهد. هر گره در این لایه، ورودی خود را از خروجی‌های متناظر در لایه‌ی قبل می‌گیرد و خروجی آن قدرت آتش[2]هر قانون می‌باشد که به صورت زیر محاسبه می‌شود:

$$y_i^{(3)} = \prod_{j=1}^{k} x_{ji}^{(3)} \qquad (3)$$

که در آن $\boldsymbol{x_{ji}^{(3)}}$، ورودی و $\boldsymbol{y_i^{(3)}}$ خروجی گره **i**ام در لایه 3ام می‌باشد و **k** تعداد خروجی لایه دوم به لایه سوم است.

لایه‌ی 4: لایه‌ی 4 لایه‌ی نرمالیزه سازی[3]می‌باشد. در این لایه هر گره ورودی خود را از تمام گره‌های لایه‌ی قبل می‌گیرد و عدد بدست‌آمده برای هر گره در لایه‌ی قبل، در این لایه نرمالیزه می‌شود. خروجی هر گره در این لایه، قدرت قانون نرمالیزه شده‌ی گره متناظر لایه‌ی قبل می‌باشد. قدرت آتش نرمالیزه شده‌ی هر قانون، از تقسیم قدرت آتش هر قانون به جمع کل قدرت آتش قانون‌ها به دست می‌آید. قدرت قانون نرمالیزه شده میزان نفوذ هر قانون در خروجی شبکه را نشان می‌دهد.

$$y_i^{(4)} = \frac{x_{ii}^{(4)}}{\sum_{j=1}^{n} x_{ji}^{(4)}} = \frac{\mu_i}{\sum_{j=1}^{n} \mu_j} \qquad (4)$$

در معادله‌ی بالا، $\mathbf{x_{ji}^{(4)}}$ خارج شده از گره **j** در لایه سوم به ورودی گروه **i** در لایه چهارم است، $\boldsymbol{y_i^{(4)}}$ خروجی گره **i**ام در لایه چهارم است و **n** تعداد کل قانونها می باشد. و μ همان قدرت آتش هر گره است که در لایه‌ی قبل محاسبه شده است و ورودی لایه‌ی چهارم به حساب می‌آید.

لایه‌ی 5: این لایه، لایه‌ی غیرفازی سازی نام دارد. هر گره در این لایه، ورودی خود را از گره متناظر در لایه‌ی قبل می‌گیرد. مقدار هر گره در این لایه با فرض مرتبه صفر بودن عبارتست از:

1 bell
2 Fire Strength
3 Normalize

$$y_i^{(5)} = x_i^{(5)}[k_{i0}] = \overline{\mu_i}[k_{i0}] \qquad (5)$$

و با فرض مرتبه‌ی 1 بودن انفیس با در نظر گرفتن دو ورودی **x1** و **x2** از رابطه‌ی زیر به دست می‌آید:

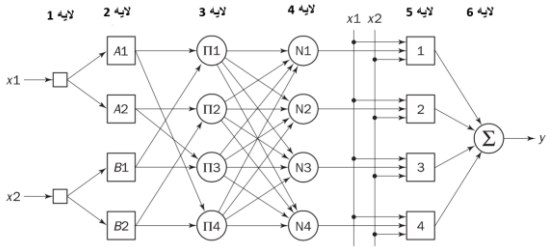
$$\mathbf{y_i^{(5)} = x_i^{(5)}[k_{i0} + k_{i1}x1 + k_{i2}x2]} \qquad (6)$$
$$\mathbf{= \overline{\mu_i}[k_{i0} + k_{i1}x1 + k_{i2}x2]}$$

که در این رابطه، **x** مقدار ورودی به هر گره، **y** خروجی گره و $\mathbf{k_{ij}}$ ها پارامترهای مربوط به قانون μ_i می باشد. چنانچه انفیس از مرتبه‌ی صفر باشد به جای **x1** و **x2** صفر قرار داده می‌شود.

لایه‌ی 6: لایه‌ی 6 تنها دارای یک گره می‌باشد که جمع گره‌های غیرفازی شده‌ی لایه‌ی قبل را محاسبه می‌کند و رابطه‌ی آن به صورت زیر است :

$$\mathbf{y} = \sum_{i=1}^{n} \mathbf{x_i^{(6)}} \qquad (7)$$

انفیس نشان داده شده در شکل 1 در واقع از لحاظ عملکردی معادل یک مدل فازی مرتبه اول سوجینو است.



شکل 1: انفیس

بهینه سازی پارامترهای دسته‌بندی کننده‌ی عصبی – فازی

برای افزایش دقت پیش‌بینی **ANFIS** و پرهیز از قرار گرفتن در حد بهینه محلی، یادگیری پارامتر توسط الگوریتم جستجوی بهینه انجام می‌شود[24]. روش یادگیری پارامتر **ANFIS** را می توان به عنوان تنظیم پارامترهای پایه و پارامترهای متعاقب آن تشریح کرد، الگوریتم جستجوی بهینه برای آموزش پارامترهایی که شامل یک کانال رو به جلو و یک کانال معکوس است، استفاده می شود. در کانال روبه جلو پارامترهای پایه ثابت هستند و ورودی آنها از محاسبات هر پنج لایه عبور می کنند و پارامترهای متعاقب با روش حداقل مربع تشخیص داده می شوند در آن نقطه پارامترهای بعدی در علائم خطای کانال معکوس تعیین می شوند و برای به روز رسانی پارامترها به لایه 2 منتقل می شوند.

فرآیند گام به گام اجرای الگوریتم جستجوی بهینه به این صورت است:

مرحله 1: راه اندازی مشکل و پارامتر قابل تطبیق

مسئله بهینه سازی، متغیرهای تصمیم گیری و محدودیت ها شرح داده شده است.

مرحله 2: مکان و حافظه عامل را راه‌اندازی می کند.

مرحله 3: محاسبه تابع کیفیت

مرحله 4: مکان جدید ایجاد می کند

مرحله 5: امکان مکان جدید را بررسی می کند

اگر مکان جدید امکان پذیر بود، عامل مکان خود را ارتقا می دهد.

در غیر این صورت، عامل در مکان فعلی می ماند و به مکان جدید ایجاد شده نمی رود.

مرحله 6: محاسبه تابع هدف مکان جدید

مقدار تابع هدف برای مکان جدید هر عامل ارزیابی شد.

مرحله 7: حافظه را به روز می کند

مرحله 8: معیارهای خاتمه را تأیید می کند

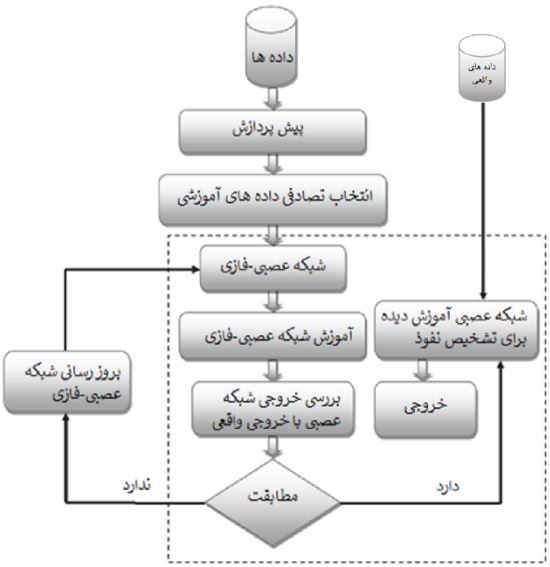
مراحل 4 تا 7 تا زمانی که حداکثر تکرار مشخص شده، انجام شود، ادامه می یابد.

هنگامی که سطح پایان به دست آمد، بهترین مکان حافظه نسبت به مقدار تابع هدف به عنوان راه حل مسئله در نظر گرفته شد.

پیاده سازی و نتایج آماری

همانطور که در بخش های قبلی اشاره کردیم 41 ویژگی برای هر اتصال در مجموعه‌داده‌های NSLKDD وجود دارد. این ویژگی‌ها دارای سه فرم پیوسته، گسسته و نمادین[1]هستند. فهرست کاملی از نام و دامنه‌ی مقادیر هر یک از ویژگی‌ها در ضمیمه 1 آمده است. طبقه‌بندی‌کننده‌ی مورد استفاده‌ی ما نمی‌تواند داده‌ها را به این صورت پردازش کند. چرا که شبکه‌ی عصبی – فازی مورد استفاده در این تحقیق توانایی استفاده از داده‌های نمادین را ندارد. بنابراین قبل از ساختن مدل نیاز به پیش پردازشی روی داده‌ها داریم که این پیش پردازش در حقیقت تبدیل مقادیر نمادین به مقادیر عددی است. در این پیش پردازش ویژگی‌های نمادین که شامل انواع پروتکل‌ها، خدمات[2]و پرچم‌ها[3]ست به مقادیر عددی در بازه‌ی 0 تا **N-1** نگاشت می شوند که در آن **N** تعداد نمادهای هر ویژگی نمادین است. به عنوان مثال ویژگی Protocol_type با سه نماد UDP،TCP و ICMP به سه عدد **0**، 1 و 2 نگاشت می‌شود. بقیه ویژگی‌ها به همان صورت اولیه استفاده می‌شوند. علاوه بر این، هر اتصال بسته به نوع خود که عادی یا نفوذ می‌باشد، با دو کلاس 0 و 1 برچسب می‌خورد. سپس 20000 داده‌ی تصادفی از مجموعه‌ی داده‌های آموزشی NSLKDD به عنوان داده‌های آموزشی و 3188 داده‌ی تصادفی از همین مجموعه داده‌ها به عنوان داده‌های بررسی[4]انتخاب شدند که هیچ اشتراکی با داده‌های قسمت اول ندارند.

ایده‌ی اصلی برای استفاده از داده‌های بررسی به منظور معتبرسازی مدل، این است که بعد از مرحله‌ی خاصی از آموزش امکان بوجود آمدن یادگیری بیش از اندازه[5]وجود دارد. اگر این آموزش بیش از اندازه اتفاق بیافتد، نمی‌توان انتظار داشت که سیستم، نسبت به داده‌های آزمایشی که جدا از داده‌های آموزش هستند به خوبی پاسخ دهد. بنابراین برای جلوگیری از آموزش بیش از اندازه‌ی داده‌ها، از مجموعه‌داده‌های بررسی استفاده می‌کنیم. پس از اتمام فرایند آموزش، سیستم عصبی – فازی تطبیق‌پذیر با مینیمم خطا برای داده‌های بررسی به عنوان سیستم نهایی انتخاب و شروع به طبقه‌بندی می‌کنیم. شمای کلی روش پیشنهادی در شکل 2 آمده است.



شکل2. شمای کلی روش پیشنهادی

برای تولید سیستم فازی اولیه، از روش خوشه‌بندی کاهشی با شعاع همسایگی **0.5** استفاده شده‌است. برای این کار، از تابع genfis2 که تابعی از برنامه‌ی متلب برای تولید قوانین فازی به روش خوشه‌بندی کاهشی است، استفاده می‌شود. پارامترهای این تابع، مجموعه‌ی داده‌های آموزشی ورودی، خروجی و شعاع همسایگی است. همانطور که ذکر شد از 20000 داده‌ی آموزشی برای آموزش دسته‌بندی کننده

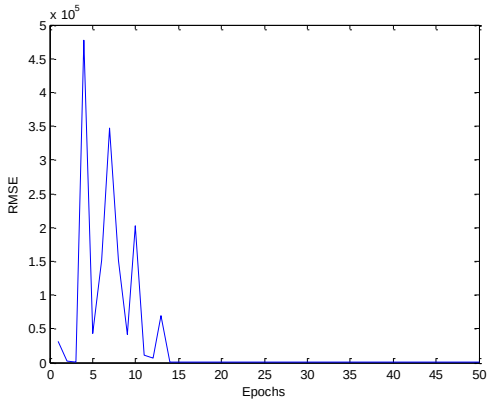
1symbolic
2services
3flags

4Validation Data
5Over Fitting

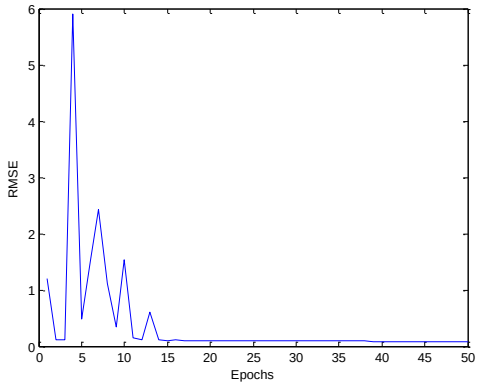
استفاده خواهد شد. ابتدا توسط این 20000 مجموعه داده برای تولید قوانین فازی استفاده خواهد شد. این مجموعه‌ی 20000 تایی، شامل 20000 رکورد داده است. هر رکورد نشان‌دهنده‌ی یک اتصال است و با 41 ویژگی مشخص می‌گردد. یعنی 20000 ردیف داده که هر ردیف شامل 41 ویژگی است. همچنین برای هر اتصال یک عدد وجود دارد که نشان دهنده‌ی نوع اتصال است. نوع اتصال می‌تواند عادی و یا نفوذ باشد که اتصال عادی با عدد 0 و اتصال نفوذ با عدد 1 نشان‌داده می‌شود. در ضمیمه 2 انواع حملات و تعداد آنها در مجموعه داده NSLKDD بصورت کامل آمده است. بنابراین در مجموع هر ردیف شامل 42 ستون است که 41 ستون اول ویژگی‌های اتصال را نشان می‌دهند و ستون 42 نوع اتصال را مشخص می‌کند. تابع genfis2، 41 ستون اول از 20000 اتصال را به عنوان پارامتر اول، ستون 42 داده‌ها یعنی نوع اتصال را به عنوان پارامتر دوم(20000 ردیف شامل ستون 42 داده‌ها) و شعاع همسایگی که عددی بین 0 و 1 است را به عنوان پارامتر سوم دریافت می‌کند. خروجی این تابع، سیستم فازی است که قوانین آن بر اساس خوشه‌بندی کاهشی تولید شده‌اند. تعداد قوانین فازی و تعداد توابع عضویت تولید شده به ازای هر ورودی، 7 است. همینطور تعداد خوشه‌ها نیز برابر 7 هستند.

برای تنظیم بیشتر و تطبیق دادن توابع عضویت با داده‌های آموزشی از مجموعه‌داده‌های آموزشی همراه با داده‌های بررسی مورد نظر، شبکه انفیس را در 50 دوره[1]آموزش دادیم. ساختار انفیسی که برای آموزش استفاده شد 632 گره داشت که مجموعا 868 پارامتر داشت که 294 پارامتر خطی و 574 پارامتر غیرخطی بودند.

مجذور میانگین مربع خطا (RMSE[2]) ، بعد از 50 دوره آموزش به ترتیب 0/0822715 برای داده‌های آموزشی و 0/0771951 برای داده‌های بررسی به دست آمد. شکل‌های 3 و 4 مقدار RMSE را به صورت تابعی از دوره برای داده‌های آموزشی و داده‌های آزمایشی به ترتیب نشان می‌دهند.



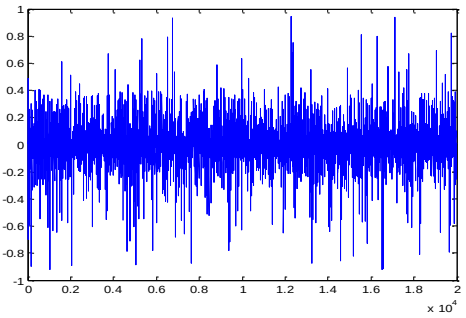
شکل 3: میزان خطا به ازای دوره‌های آموزش برای داده‌های آموزشی



شکل 4: میزان خطا به ازای دوره‌های آموزش برای داده‌های آزمایش

اغلب سعی بر آن است تا با آموزش یک شبکه بتوان مقادیر مناسبی از خروجی را برای داده‌های ورودی بدست آورد و این عمل همیشه به صورت 100 درصد امکان پذیر نیست و هر شبکه‌ی آموزش دیده، به طور معمول مقداری خطا در تخمین خروجی مورد انتظار دارا می باشد. شکل 4 تفاوت مقدار واقعی و مقدار بدست آمده از خروجی انفیس را برای داده‌های آموزشی بعد از 50 دوره آموزش نشان می دهد.

1 Epoch
2 Root Mean Squared Error

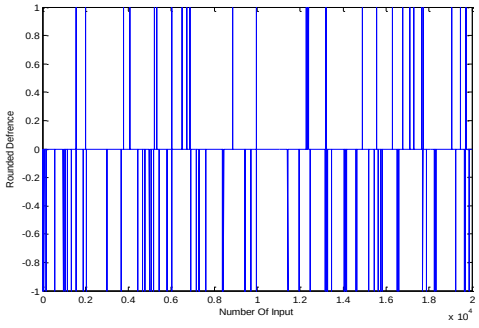


شکل 5: تفاوت مقدار واقعی و مقدار بدست آمده از خروجی انفیس برای داده-های آموزشی

همانطور که از شکل 5 می توان استنتاج نمود خروجی شبکه‌ی انفیس که در اصل شماره کلاس داده‌ی ورودی می باشد (0 برای اتصال های عادی و 1 برای اتصال‌های نفوذ) یک مقدار عددی صحیح نمی باشد. بنابراین ما نیاز داریم مقدار خروجی بدست آمده را برای بدست آوردن شماره کلاس گرد کنیم [20]. در این شکل محور افقی تعداد تکرار و محور عمودی میزان اختلاف مقدار واقعی و مقدار بدست آمده از خروجی انفیس را نشان میدهد.

ایده‌ی اصلی ما برای محاسبه‌ی تفاوت مقدار واقعی و مقادیر خروجی انفیس این است که اگر خروجی درست پیش بینی شده باشد مقدار حاصل از تفاوت باید صفر باشد. یعنی اگر فرض کنیم یک اتصال عادی باشد و مقدار 0 داشته باشد، درصورت تشخیص درست نیز مقدار صفر خواهد داشت و حاصل اختلاف این دو صفر است. همینطور در صورتی که اتصال از نوع نفوذ باشد مقدار 1 خواهد گرفت و در صورت تشخیص درست نیز مقدار 1 می‌گیرد که باز هم حاصل اختلاف صفر می‌شود.

پس اگر پیش‌بینی درست انجام شده باشد حاصل تفاوت مقدار واقعی اتصال و مقدار بدست‌آمده توسط انفیس، باید صفر باشد. برای مقادیر غیرصحیح از تکنیک گرد کردن استفاده می‌کنیم. بدین صورت که فرضا یک اتصال عادی باشد، بنابراین مقدار خروجی واقعی برابر 0 است. حال اگر مقدار بدست آمده از انفیس عددی بین 0 و 0/5 باشد، حاصل تفاوت مقدار اصلی و مقدار بدست‌آمده توسط انفیس، عددی بین صفر و 0/5- خواهد بود. همچنین چنانچه خروجی حاصل از انفیس بین 0 تا 0/5- باشد حاصل تفاوت مقدار اصلی و مقدار بدست‌آمده توسط انفیس، عددی بین 0 تا 0/5 است. این تفاوت را به صفر گرد می‌کنیم. همچنین اگر حاصل تفاوت مقدار واقعی و مقدار بدست‌آمده توسط انفیس، عددی بیشتر از 0/5 یا کوچکتر از 0/5- باشد آن را به عدد صحیح غیرصفر گرد می‌کنیم و بعنوان پیش‌بینی اشتباه ثبت می‌کنیم. بر این اساس پس از عملیات گرد کردن، اعداد صفر نشان دهنده‌ی پیش-بینی درست و اعداد غیرصفر نشان دهنده‌ی پیش‌بینی غلط هستند، برای پیدا کردن حد آستانه از آزمایش و خطا استفاده شد تا مقدار مناسب برای این مقدار مشخص شود. که استفاده از روشهای دیگر و مقادیر دیگر باعث میشد که نرخ تشخیص خطا و نرخ هشدار غلط به اندازه مطلوب نباشد. در شکل 6 نمودار تفاوت خروجی واقعی و خروجی انفیس را به صورت گردشده نشان داده شده است. بر اساس نتایج بدست‌آمده از آموزش انفیس و آزمایش دسته-بندی کننده‌ی عصبی – فازی با داده‌های آموزشی که در شکل 6 ارائه شده است، از کل 20000 داده‌ی آموزشی، تعداد تشخیص‌های درست برابر 19910 عدد بوده است. یعنی تعداد صفرهای بدست‌آمده، حاصل از اختلاف مقدار واقعی اتصال و مقدار بدست‌آمده توسط انفیس، برابر 19910 تا از 20000 اتصال است. این یعنی 99/55 درصد از اتصالات به درستی تشخیص داده‌شده‌اند. این تشخیص شامل تشخیص درست داده‌های عادی و نفوذ است. یعنی از 20000 اتصال که شامل اتصالات عادی و نفوذ است، 99/55 درصد، به درستی تشخیص داده شده‌اند. از بین این 20000 اتصال، 10693 اتصال عادی و 9307 اتصال نفوذ بوده‌اند. در الگوریتم 1، مراحل انجام کار و نحوه شمارش تعداد تشخیص درست و نادرست ارائه شده است.



شکل 6: مقدار گرد شده‌ی تفاوت خروجی واقعی و خروجی انفیس

الگوریتم 1. سیستم تشخیص نفوذ پیشنهادی مبتنی بر انفیس
{ تعیین تشخیص صحیح ترافیک توسط انفیس}

```
Input Dataset
Preprocess Dataset  //function
Select random Dataset
Construct ANFIS Classifier with train and test dataset
//function
Lx: Output of the classification
Fx:  Real Output in Dataset (feature of 42)
begin
  Sx:  Lx – Fx
  if  -0.5 <= Sx <= 0.5  then
       True Detection
  else
       False Detection
  end
 end
```

معیارهای ارزیابی

معیارهای استاندارد گوناگونی برای ارزیابی سیستم های تشخیص نفوذ ارائه شده است. از جمله می توان به نرخ تشخیص خطا[1]و نرخ هشدارهای غلط[2]اشاره نمود. نرخ تشخیص خطا از تقسیم تعداد حملاتی که به درستی تشخیص داده شده‌اند بر کل تعداد حملات بدست می‌آید و نرخ هشدارهای غلط در حقیقت نسبت تعداد اتصالات عادی که به اشتباه به عنوان نفوذ تشخیص داده شده‌اند به کل تعداد اتصالات عادی است. در ادامه معیارهای ارزیابی آورده شده است:

$$\text{Accuracy}=\frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (8)$$

$$\text{Recall (TPR)}=\frac{(TP)}{(TP+FN)} \quad (9)$$

$$\text{FPR}=\frac{(FP)}{(TN+FP)} \quad (10)$$

$$\text{F\_measure} =\frac{2}{\frac{1}{Recall}+\frac{1}{Precision}} \quad (11)$$

که:

True Positive (TP): تعداد نمونه های نفوذ که به صورت نفوذ طبقه بندی شده اند.

True Negative (TN): تعداد نمونه های نرمال که به صورت نرمال طبقه بندی شده اند.

False Positive (FP): تعداد نمونه های نرمال که به صورت نفوذ طبقه بندی شده اند.

False Negative (FN): تعداد نمونه های نفوذ که به صورت نرمال طبقه بندی شده اند.

در جدول 1 مدل پیشنهادی را با چهار معیار ذکر شده در رابطه‌های 8 تا 11 با داده های آموزشی و آزمایشی مورد بررسی قرار داده ایم و در ادامه مدل پیشنهادی را با استفاده از معیارهای نظیر، "نرخ تشخیص خطا" و "نرخ هشدار غلط"، که کارایی سیستم‌های تشخیص نفوذ طراحی شده را بهتر نشان می‌دهند[25]، با مدل‌های دیگر مقایسه کردیم.

برای نرخ تشخیص خطا، هرچه این نرخ به 1 نزدیکتر باشد کارایی بهتر را نشان میدهد و هرچه نرخ هشدار غلط کمتر باشد (به صفر نزدیکتر باشد) بهتر است.

برای آزمایش دسته‌بندی کننده با داده‌های آموزشی، داده-های آموزشی مورد استفاده برای آموزش، به 4 دسته تقسیم، و نتایج ارائه شده در جدول 1 میانگین نتایج بدست آمده از آزمایش با 4 دسته‌داده‌ی آموزشی است. نتایج این جدول نشان می دهد که طبقه‌بندی‌کننده مورد نظر در کلاس‌بندی داده‌های آموزشی که بر اساس آنها آموزش دیده است، بسیار خوب عمل می کند. برای درک بهتر موضوع، 9307

1 True Positive Rate (TPR)

2 False Positive Rate (FPR)

اتصال از 20000 اتصالی که برای آموزش استفاده شده بودند از نوع نفوذ بوده‌اند. در هنگام آزمایش دسته‌بندی کننده با داده‌های آموزشی، 9281 اتصال از 9307 اتصال نفوذ، توسط دسته‌بندی کننده به درستی از نوع نفوذ تشخیص داده شدند. برای نرخ هشدار غلط نیز بر اساس تعریف، مقدار آن محاسبه و در جدول 1 ارائه شده‌است.

به منظور بررسی بیشتر کارایی طبقه‌بندی‌کننده‌ی ارائه شده و بررسی عملکرد طبقه‌بندی‌کننده‌ی عصبی - فازی در قبال داده‌هایی که در فرآیند آموزش مورد استفاده قرار نگرفته‌اند (داده‌های آزمایش)، 4075 داده‌ی تصادفی دیگر از مجموعه‌داده‌های آزمایش NSLKDD به منظور ارزیابی سیستم انتخاب شده‌اند که با داده‌های آموزشی و آزمایشی هیچ اشتراکی ندارند. نتیجه‌ی آزمایش دسته‌بندی کننده با این داده‌ها در جدول 1 ارائه شده است. در این جدول کارایی طبقه‌بندی‌کننده‌ی ارائه شده را بر اساس آزمایش با 4075 داده که در فاز آموزش استفاده نشده‌اند و جزء داده-های آزمایش مجموعه داده NSLKDD هستند، نشان می-دهد.

جدول 1: ارزیابی مدل پیشنهادی با داده‌های آموزشی و آزمایشی

نوع ارزیابی	داده های آموزش	داده های آزمایش
Accuracy	99/81	۹۵/۷۹
TPR	۹۹/۷۲	۹۵/۶۵
FPR	۰/۰۶	۰/۲۱۹
F_measure	99/76	۹۵/۷۳

همچنین برای بررسی بیشتر طبقه‌بندی‌کننده‌ی عصبی – فازی ارائه شده برای سیستم تشخیص نفوذ، نتایج حاصل از این طبقه‌بندی‌کننده، با نتایج تحقیقات دیگر که با روش های مختلف اقدام به ساخت سیستم تشخیص نفوذ کرده‌اند مقایسه شده است. هرچند که این مقایسه به علت وجود عدم دسترسی به داده‌های سایر تحقیقات در هنگام آموزش و آزمایش، نمی تواند عادلانه باشد. چرا که در سایر روش ها مشخص نیست که دقیقا از کدام اتصالات NSLKDD برای آموزش و آزمایش استفاده شده است. سعی ما بر این بوده است که داده‌های انتخاب شده در فاز آموزش و آزمایش کاملا تصادفی باشند. اما در عین حال نتایج به دست آمده، می‌تواند تا حد زیادی توانایی دسته‌بندی‌کننده عصبی – فازی تطبیق پذیر را در این مساله‌ی طبقه‌بندی نشان دهد و این اطمینان را به ما بدهد که طبقه‌بندی‌کننده‌ی ارائه شده در این بخش، به اندازه‌ی کافی توانمند است و کارایی آن خارج از حد انتظار نمی‌باشد چرا که رویکرد ما در استفاده از دسته‌بندی‌کننده‌ی عصبی – فازی این بوده است که انفیس به علت قدرت یادگیری بدون دانش اولیه و استفاده از منطق فازی و شبکه‌عصبی، باید طبقه‌بندی-کننده‌ی قدرتمندی باشد.

تحلیل و مقایسه نتایج

محققان در زمینه‌ی بهینه‌سازی سیستم‌های تشخیص نفوذ تلاش‌های زیادی کرده‌اند. روش‌های متعددی برای ساخت سیستم تشخیص نفوذ توسط محققان به کار گرفته شده است. در این قسمت ما به مقایسه‌ی نتایج حاصل از سیستم تشخیص نفوذ ساخته شده توسط دسته‌بندی‌کننده‌ی عصبی – فازی با سایر روش‌هایی که برای ساخت این سیستم مورد استفاده قرار گرفته‌اند خواهیم پرداخت.

در جدول 2 نیز به مقایسه‌ی نتایج حاصل از این تحقیق با نتایج سایر تحقیقات خواهیم پرداخت. نکته‌ای که باید به آن توجه داشت، این است که به دلیل استفاده از داده‌های تصادفی در هنگام آموزش، ممکن است که این مقایسه تا حدی عادلانه نباشد، ولی نتایج جداول برای پذیرفتن طبقه-بندی‌کننده عصبی – فازی، به عنوان یک طبقه‌بندی‌کننده‌ی مناسب، کفایت می‌کند و نشان‌دهنده‌ی این مطلب است که این طبقه‌بندی‌کننده از توانایی لازم در حد روشهای مشابه وحتی بیشتر، برخوردار است. همانطور که در جدول 2 مشخص است، نرخ تشخیص خطا و نرخ هشدار غلط

بدست‌آمده توسط این مقاله، نسبت به سایر روش‌ها، برتری قابل قبولی دارد.

برای مقایسه‌ی نتایج بدست آمده در بخش داده‌های آموزش نیز، نتایج حاصل از آموزش دسته‌بندی کننده‌ی عصبی – فازی با داده‌های آموزشی را با یک تحقیق دیگر که نتایج آن بر اساس آزمایش با داده‌های آموزشی ارائه شده است، مقایسه نموده‌ایم. در بخش آموزش این تحقیق ما تعداد 20000 داده‌ی تصادفی از مجموعه‌داده‌ی NSLKDD را انتخاب کرده بودیم و آموزش نیز بوسیله دسته‌بندی‌کننده‌ی عصبی – فازی انجام شد.

جدول 2: مقایسه‌ی نتایج آزمایش سیستم تشخیص نفوذ با سایر نتایج بر اساس معیارهای نرخ تشخیص خطا و نرخ هشدار غلط

روش ساخت سیستم تشخیص نفوذ	نرخ تشخیص خطا	نرخ هشدار غلط	مجموعه داده
CSOM+RBF[26]	95/27	1/64	NSL_KDD
Deep Nural Network[27]	94/62	0/97	NSL_KDD
TLMD [12]	93/32	0/06	KDD 99
LSSVM-IDS [13]	91/12	0/38	KDD 99, NSL_KDD, Kyoto 2006+
Fuzzy classification [۱۵]	94/4	0/35	DARPA
ESC-IDS [۲۰]	95/3	1/9	KDD 99
PNrule [۲۱]	91/1	0/4	NSL_KDD
PCA [۲۸]	93/83	0/616	KDD 99
Proposed IDS	95/56	0/219	**NSL_KDD**

پاندا[1] و همکارانش، در تحقیق خود از تعداد 25192 اتصال مجموعه‌داده‌ی NSLKDD برای آموزش مدل خود استفاده کردند[29]. آنها در تحقیق خود از یک روش ترکیبی استفاده نمودند. این روش ترکیبی شامل درخت تصمیم[2]، تجزیه و تحلیل مؤلفه های اصلی[3]، ماشین بردار پشتیبان[4]، END، Grading و جنگل تصادفی[5] می‌باشد.

این محققین، از مجموعه‌داده‌های آموزشی، برای آزمایش سیستم تشخیص نفوذ خود استفاده نموده‌اند. نتایجی که این محققان در تحقیق خود برای آزمایش با داده‌های آموزشی مورد استفاده‌ی خود بدست آورده‌اند، به همراه نتایج حاصل از این تحقیق، در جدول 3 ارائه شده‌است.

جدول 3: مقایسه‌ی نتایج این تحقیق برای آزمایش سیستم تشخیص نفوذ با داده‌های آموزشی باروش ترکیبی Panda

روش	نرخ تشخیص خطا	نرخ هشدار غلط
Hybrid Intelligent Approach	99/5	0/5
Proposed IDS	99/72	0/06

آنچه در جدول 3 مشخص است، این است که روش دسته-بندی‌کننده‌ی عصبی – فازی به تنهایی با روش ترکیبی که در بالا ذکر شد رقابت می‌کند. وقتی ما دسته‌بندی‌کننده را با 41 ویژگی آموزش می‌دهیم (در تحقیق پاندا و همکارانش نیز از 41 ویژگی استفاده شده است)، نرخ تشخیص خطا و نرخ هشدار غلط هردو نسبت به روش دیگر بهینه‌تر هستند و این خود گواهی بر قدرت این دسته-بندی‌کننده عصبی- فازی می‌باشد.

نتیجه گیری

در این مقاله، به مقایسه‌ی نتایج حاصل شده از دسته‌بندی کننده‌ی عصبی- فازی با چند تحقیق دیگر پرداختیم. ابتدا نتایج مقادیر بدست‌آمده برای نرخ تشخیص خطا و نرخ هشدار غلط حاصل از دسته‌بندی کننده که آموزش دیده بود و با داده‌های آزمایشی پایگاه‌داده‌ی NSLKDD آزمایش شده‌بود، با نتایج معیارهای سایر محققین مقایسه شد. نتایج

[1] Panda
[2] Decision Tree
[3] Principal Component Analysis
[4] Support Vector Machine
[5] Random Forest

جدول 2 حکایت از کیفیت و قدرت دسته‌بندی کننده‌ی عصبی – فازی در مقایسه با سایر نتایج دارد. همانطور که در بخش‌های قبلی نیز اشاره شد پایگاه‌داده‌ی NSLKDD، دارای دو دسته مجموعه‌داده‌ی تفکیک‌شده برای آموزش و آزمایش سیستم تشخیص نفوذ می‌باشد. این دو مجموعه-داده هیچ اشتراکی ندارند و بر این اساس می‌توان سیستم تشخیص نفوذ ساخته شده را به طور کامل محک زد. این محک توسط "مجموعه‌داده‌ی بررسی" انجام می‌شود. علاوه بر این، توسط "مجموعه‌داده‌ی آزمایش"، به آزمایش سیستم تشخیص نفوذ ساخته شده پرداخته‌ایم. در نهایت، به این نتیجه می‌رسیم که دسته‌بندی کننده‌ی عصبی – فازی مبتنی بر فازی سوجینو، می‌تواند به عنوان یک دسته‌بندی کننده‌ی قابل قبول برای سیستم تشخیص نفوذ مورد استفاده قرار گیرد.

در این تحقیق باور ما بر این بوده است که، طبقه‌بندی - کننده‌ی عصبی– فازی، به دلیل نوع ساختارش می‌تواند یک دسته‌بندی‌کننده‌ی قدرتمند باشد و همچنین نتایجی که بررسی شد قابلیت بهبود دارند. این قابلیت بهبود بستگی به عوامل زیادی دارد که از آن جمله می‌توان به استفاده از پایگاه‌داده‌ی بهتر نسبت به NSLKDD اشاره کرد.

مراجع

[1] E. Corchado and Á. Herrero, "Neural visualization of network traffic data for intrusion detection," *Applied Soft Computing,* vol. 11, no. 2, pp. 2042-2056, 2011.

[2] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of computer security,* vol. 6, no. 3, pp. 151-180, 1998.

[3] G. Folino and P. Sabatino, "Ensemble based collaborative and distributed intrusion detection systems: A survey," *Journal of Network and Computer Applications,* vol. 66, pp. 1-16, 2016.

[4] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications,* vol. 12, no. 2, pp. 493-501, 2019.

[5] S.-Y. Wu and E .J. E. S. w. A. Yen, "Data mining-based intrusion detectors," vol. 36, no. 3, pp. 5605-5612, 2009.

[6] N. Pandeeswari and G. Kumar, "Anomaly detection system in cloud environment using fuzzy clustering based ANN," *Mobile Networks and Applications,* vol. 21 ,no. 3, pp. 494-505, 2016.

[7] I. Škrjanc, J. A. Iglesias, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide, "Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey," *Information Sciences,* vol. 4 ,90pp. 344-368, 2019.

[8] A. Abraham and R. Jain, "Soft computing models for network intrusion detection systems," in *Classification and clustering for knowledge discovery*: Springer, 2005, pp. 191-207.

[9] K. Faraoun and A. Boukelif, "Neural networks learning improvement using the K-means clustering algorithm to detect network intrusions," *INFOCOMP Journal of Computer Science,* vol. 5, no. 3, pp. 28-36, 2006.

[10] T. Dash, "A study on intrusion detection using neural networks trained with evolutionary algorithms," *Soft Computing,* vol. 21, no. 10, pp. 2687-2700, 2017.

[11] R. S. K. Jonnalagadda and R. P. Reddy, "A literature survey and comprehensive study of intrusion detection," *International Journal of Computer Applications,* vol. 81, no. 16, pp. 40-47, 2.013

[12] Y. Yuan, L. Huo, and D. Hogrefe, "Two layers multi-class detection method for network intrusion detection system," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, 2017: IEEE, pp. 767-772 .

[13] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE transactions*

transactions on systems, man, and cybernetics, vol. 23, no. 3, pp. 665-685, 1993.

[24] A. G. Hussien *et al.*, "Crow Search Algorithm: Theory, Recent Advances, and Applications," *IEEE Access,* vol. 8, pp. 173548-173565, 2020, doi: 10.1109/ACCESS.2020.3024108.

[25] Y. Zhou, G. Cheng, S. Jiang, and M. Dai , "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks,* vol. 174, p. 107247, 2020.

[26] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, and A. Razaque, "Cascaded hybrid intrusion detection model based on SOM and RBF neural networks," *Concurrency and Computation: Practice and Experience,* vol. 32, no. 21, p. e5233, 2020.

[27] M. M. U. Chowdhury, F. Hammond, G. Konowicz, C. Xin, H. Wu, and J. Li, "A few-shot deep learning approach for improved intrusion detection," in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, 2017: IEEE, pp. 456-462 .

[28] R. Beghdad, "Critical study of neural networks in detecting intrusions," *Computers & security,* vol. 27 ,no. 5-6, pp. 168-175, 2008.

[29] M. Panda, A. Abraham, and M. R. Patra, "Hybrid intelligent systems for detecting network intrusions," *Security and Communication Networks,* vol. 8, no. 16, pp. 2741-2749, 2015.

on computers, vol. 65, no. 10, pp. 2986-2998, 2016.

[14] R. Beghdad, "Critical study of supervised learning techniques in predicting attacks," *Information Security Journal: A Global Perspective,* vol. 19, no. 1, pp. 22-35, 2010.

[15] J. Gomez and D. Dasgupta, "Evolving fuzzy classifiers for intrusion detection," in *Proceedings of the 2002 IEEE Workshop on Information Assurance*, 2002, vol ,6 .no. 3: New York: IEEE Computer Press, pp. 321-323 .

[16] C. Jirapummin, N. Wattanapongsakorn, and P. Kanthamanon, "Hybrid neural networks for intrusion detection system," in *Proc. of ITC–CSCC*, 2002, vol. 7, pp. 928-931 .

[17] T. Horeis, "Intrusion detection with neural networks–combination of self-organizing maps and radial basis function networks for human expert integration," *Computational Intelligence Society Student Research Grants,* 2003.

[18] S.-J. Han and S.-B. Cho, "Evolutionary neural networks for anomaly detection based on the behavior of a program," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 36, no. 3, pp. 559-570, 2005.

[19] Y. Chen, A. Abraham, and B. Yang, "Hybrid flexible neural-tree-based intrusion detection systems," *International journal of intelligent systems,* vol. 22, no. 4, pp. 337-352, 2007.

[20] A. N. Toosi and M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers," *Computer communications,* vol. 30, no. 10, pp. 2201-2212, 2007.

[21] S. Ghosh, A. Pal, A. Nag, S. Sadhu, and R. Pati, "Network anomaly detection using a fuzzy rule-based classifier," *Computer, Communication and Electrical Technology,* pp. 61-65, 2017.

[22] M. Negnevitsky and A. Intelligence, "A guide to intelligent systems," *Artificial Intelligence, 2nd edition, pearson Education,* 2005.

[23] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE*

ضمیمه ۱

جدول لیست کامل ویژگی‌های تعریف شده برای هر داده‌ی اتصال

برچسب	ویژگی‌های شبکه	توضیحات	بازه‌ی مقادیر
A	Duration	طول اتصال بر حسب ثانیه	21-0
B	protocol_type	نوع پرتوکل به عنوان مثال tcp, udp و ...	Tcp – udp – icmp
C	Service	سرویس شبکه در مقصد، به‌عنوان مثال http، telnet، ...	http , talent , ftp_data , private , remote_job , mtp , eco_i ,suodup
D	Flag	حالت اتصال به شبکه	SF , S0,S1,S2,S3,REJ,SH,PSTR,…
E	src_bytes	تعداد بایت‌های یک داده ارسالی از مبدا به مقصد	1379963888-0
F	dst_bytes	تعداد بایت‌های ارسالی از مقصد به مبدا	1309937401-0
G	Land	1 اگر اتصال از/به host/port یکسان، در غیر این صورت 0	1-0
H	wrong_fragment	تعداد قطعات اشتباه	3-0
I	Urgent	تعداد بسته‌های ضروری	3-0
J	Hot	تعداد شاخص‌های hot	77-0
K	num_failed_logins	تعداد تلاش‌های انجام شده برای ورود	5-0
L	logged_in	1 اگر ورود موفق داشته‌باشد; در غیر اینصورت 0	1-0
M	num_compromised	تعداد اتصالات در معرض خطر	7479-0
N	root_shell	1اگر پوشش ریشه بدست آمده باشد; در غیر اینصورت 0	1-0
O	su_attempted	1 اگر تلاش برای اجرای دستور su root انجام شده باشد در غیر اینصورت0	1-0
P	num_root	تعداد دسترسی به ریشه	7468-0
Q	num_file_creations	تعداد عملیات ایجاد فایل	43-0
R	num_shells	تعداد ایجاد دستورات sell	2-0
S	num_access_files	تعداد دستورات دسترسی به فایل	9-0
T	num_outbound_cmds	تعداد دستورات خارج از محدوده برای دسترسی به نشست ftp	0
U	is_host_login	1 اگر ورود متعلق به hot لیست باشد،درغیر اینصورت 0	1-0
V	is_guest_login	1 اگر کاربر بعنوان مهمان وارد شود; درغیر اینصورت 0	1-0
W	Count	تعداد اتصال به یک میزبان در اتصالات اخیر در 2 ثانیه گذشته	511-0
X	srv_count	تعداد اتصالاتی که سرویس یکسان در اتصال اخیر در بیشتر از 2 ثانیه دارند	511-0
Y	serror_rate	درصد اتصالاتی که خطای SYN دارد	1-0
Z	srv_serror_rate	درصد اتصالاتی که خطای SYN دارد	1-0
AA	rerror_rate	در صد اتصالاتی که خطای REJ دارد	1-0
AB	srv_rerror_rate	در صد اتصالاتی که خطای REJ دارد	1-0
AC	same_srv_rate	درصد اتصالاتی که سرویس یکسان دارند	1-0
AD	diff_srv_rate	درصد اتصالاتی که سرویس مختلف دارند	1-0
AE	srv_diff_host_rate	درصد اتصالاتی که میزبان مختلف دارند	1-0
AF	dst_host_count	تعداد اتصالات از یک میزبان به مقصد در طی یک زمان مشخص	255-0
AG	dst_host_srv_count	تعداد اتصالات از یک میزبان به مقصد برای دستیابی به سرویس	255-0
AH	dst_host_same_srv_rate	درصد اتصالات از یک میزبان به مقصد برای دستیابی به یک سرویس یکسان	1-0
AI	dst_host_diff_srv_rate	درصد اتصالات از یک میزبان به مقصد برای دستیابی به سرویس های مختلف	1-0
AJ	dst_host_same_src_port_rate	درصد اتصالات از یک میزبان به مقصد از یک پورت یکسان	1-0
AK	dst_host_srv_diff_host_rate	درصد اتصالات از یک میزبان به مقصد های مختلف برای دستبابی به سرویس	1-0
AL	dst_host_serror_rate	درصد اتصالاتی از یک میزبان به یک مقصد مشخص که خطای SYN دارند	1-0
AM	dst_host_srv_serror_rate	درصد اتصالات از یک میزبان با یک سرویس مشخص به مقصد که خطای SYN دارند	1-0
AN	dst_host_rerror_rate	درصد اتصالاتی از یک میزبان به یک مقصد مشخص که خطای REJ دارند	1-0
AO	dst_host_srv_rerror_rate	درصد اتصالات از یک میزبان با یک سرویس مشخص به مقصد که خطای REJ دارند	1-0

ضمیمه ۲

جدول توزیع حملات

Class	Training			Testing		
	Attack names	Samples	Total	Attack names	Samples	Total
DOS	teardrop	979	391,458	Apache 2	794	229,853
	smurf	280,790		Back	1098	
	neptune	107,201		land	9	
	Pod	264		mailbomb	5000	
	Back	2203		neptune	58,001	
	Land	21		pod	87	
				processtable	759	
				smurf	164,091	
				teardrop	12	
				udpstorm	2	
Probe	satan	1589	4107	ipsweep	306	4166
	nmap	231		mscan	1053	
	ipsweep	1247		nmap	84	
	portsweep	1040		portsweep	354	
				saint	736	
				satan	1633	
U2R	perl	3	52	buffer overflow	22	70
	buffer overflow	30		loadmodule	2	
	rootkit	10		perl	2	
	loadmodule	9		ps	16	
				rootkit	13	
				sqlattack	2	
				xterm	13	
R2L	ftp write	8	1126	ftp write	3	16347
	Warezclient	1020		guess passwd	4367	
	Warezmaster	20		imap	1	
	Spy	2		multihop	18	
	guess passwd	53		named	17	
	Imap	12		phf	2	
	multihop	7		sendmail	17	
	phf	4		snmpgetattack	7741	
				Snmpguess	2406	
				warezmaster	1602	
				worm	2	
				xlock	9	
				xsnoop	4	
				httptunnel	158	

حملاتی که به صورت خاکستری مشخص شده است، حملاتی هستند که در داده های آزمایشی وجود دارد و در داده های آموزشی وجود نداشته است.