Journal of Electronical & Cyber Defence
Vol. 9, No. 1, 2021, Serial No. 33

Detecting of Botnets' Malicious Domains with Deep Autoencoder Neural Network

M. Asadi, S. Parsa*, V. Vosoghi

*Iran University of Science and Technology, Tehran, Iran

(Received: 02/04/2020, Accepted: 05/08/2020)

ABSTRACT

Botnet is a group of hosts infected with the same malicious code and managed by an attacker or Botmaster through one or more command and control (C&C) servers. The new generation of Botnets generates C&C domain name server's list dynamically. This dynamic list created by a domain generation algorithm helps an attacker to periodically change its C&C servers and prevent their addresses from being blacklisted. Each infected host generates a large number of domain names using a predefined algorithm and attempts to map them to their corresponding addresses by sending queries to the domain server. In this paper, the deep autoencoder neural network is used to identify domains without any knowledge of their generating algorithm, and the performance of the proposed method is compared with the performance of machine learning algorithms. Initially, a new dataset is created by combining a data set with normal domains and two datasets containing malicious and abnormal domains and both manual and automated methods are used to extract the features of the new dataset. Deep autoencoder neural network is applied to new and pre-processed datasets and the results are compared with machine learning algorithms. Based on the obtained results, it is possible to identify the malicious domains generated by domain generating algorithms using the deep autoencoder neural network with a higher speed and an accuracy rate larger than 98.61%.

Keywords: Botnet, Domain Generation Algorithms (DGAs), Feature Extraction, Deep Neural Network, Deep Autoencoder Neural Network

* Corresponding Author Email: Parsa@iust.ac.ir

نشریه علمی "پدافند الکترونیکی و سایبری"
سال نهم، شماره ۱، بهار ۱۴۰۰، ص ۷۴-۶۱

علمی – پژوهشی

شناسایی دامنه‌های بدخواه شبکه‌های بات با استفاده از شبکه عصبی خود رمزگذار عمیق

مهدی اسدی[۱]، سعید پارسا[۲]، وحید وثوقی[۳]

۱- مربی، گروه مهندسی کامپیوتر، واحد خامنه، دانشگاه آزاد اسلامی، خامنه، ایران، ۲- دانشیار، گروه مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران، ۳- کارشناس ارشد، گروه مهندسی کامپیوتر، واحد شبستر، دانشگاه آزاد اسلامی، شبستر، ایران

(دریافت: ۱۳۹۹/۰۱/۱۴، پذیرش: ۱۳۹۹/۰۵/۱۵)

چکیده

هر شبکه بات گروهی از میزبان‌هایی است که با کد بدخواه یکسانی آلوده‌شده و از طریق یک یا چند سرویس‌دهنده فرمان و کنترل توسط مهاجم یا مدیر بات هدایت می‌شوند. در شبکه‌های بات نسل جدید فهرست نام‌های دامنه سرویس‌دهنده‌های فرمان و کنترل به‌صورت پویا ایجاد می‌شود. این فهرست پویا که توسط یک الگوریتم تولید دامنه ایجاد می‌شود به مهاجم کمک می‌کند تا مکان سرویس‌دهنده‌های فرمان و کنترل خود را به‌صورت دوره‌ای تغییر داده و از قرار گرفتن آدرس‌های آن‌ها در فهرست‌های سیاه جلوگیری کند. هر میزبان آلوده با استفاده از یک الگوریتم از پیش تعریف‌شده، تعداد زیادی نام دامنه تولید کرده و با ارسال پرس‌وجوهای سرویس‌دهنده دامنه تلاش می‌کند آن‌ها را به آدرس‌های متناظرشان نگاشت کند. در این مقاله، از الگوریتم شبکه عصبی خود رمزگذار عمیق برای شناسایی دامنه‌هایی که هیچ‌گونه آگاهی از الگوریتم تولید آن‌ها وجود نداشته است، استفاده‌شده و عملکرد روش پیشنهادی با عملکرد الگوریتم‌های یادگیری ماشین مقایسه شده است. ابتدا مجموعه داده جدیدی از ترکیب یک مجموعه داده با دامنه‌های سالم و دو مجموعه داده حاوی دامنه‌های بدخواه و ناسالم ایجادشده و از دو سناریوی دستی و خودکار برای استخراج ویژگی‌های مجموعه داده جدید استفاده‌شده است. شبکه عصبی خود رمزگذار عمیق بر روی مجموعه داده جدید و پیش‌پردازش شده اعمال‌شده و نتایج در مقایسه با الگوریتم‌های یادگیری ماشین بررسی‌شده است. با توجه به نتایج به‌دست‌آمده، می‌توان با استفاده از شبکه عصبی خود رمزگذار عمیق، دامنه‌های بدخواه تولیدشده توسط الگوریتم‌های تولید دامنه را با سرعت بیشتر و نرخ صحت بیشتر از ۹۸,۶۱٪ شناسایی کرد.

کلیدواژه‌ها: شبکه بات، الگوریتم‌های تولید دامنه، استخراج ویژگی، شبکه عصبی عمیق، شبکه عصبی خود رمزگذار عمیق

۱- مقدمه

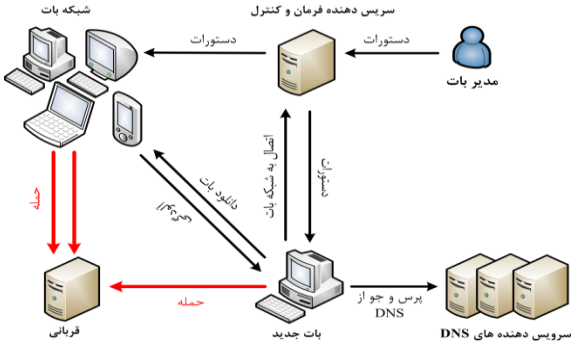
شبکه بات، شبکه‌ای متشکل از رایانه‌های آلوده به نام بات است و تمام این بات‌ها توسط یک مدیر بات[۱] کنترل می‌شوند [۱]. شمای کلی یک شبکه بات در شکل (۱) نشان داده شده است. الزامی به وجود تمامی بات‌ها در یک شبکه نیست، بلکه بات‌ها می‌توانند در شبکه‌های مختلف توزیع شوند. تمام بات‌ها از کانال فرمان و کنترل[۲] استفاده می‌کنند و توسط مدیر بات کنترل می‌شوند، بنابراین حملات آن‌ها می‌تواند به‌خوبی طراحی‌شده و هم‌زمان اجرا شود. از شبکه‌های بات می‌توان برای انجام حملات بسیاری ازجمله ارسال هرزنامه‌ها، سرقت اطلاعات شخصی و انتشار حملات انکار سرویس توزیع‌شده استفاده کرد. امروزه اغلب بات‌ها بر استفاده از الگوریتم‌های تولید دامنه تمرکز دارند تا فهرستی از نام‌های دامنه را برای اتصال با سرویس‌دهنده فرمان و کنترل خود ایجاد کنند. بر اساس مطالعات اخیر، حدود ده درصد از تمامی رایانه‌های موجود بر روی اینترنت توسط شبکه‌های بات آلوده‌شده‌اند. در میان نرم‌افزارهای مخرب و بدخواه، شبکه‌های بات گسترده‌ترین و جدی‌ترین تهدیدی هستند که امروزه به‌طورمعمول در حملات سایبری رخ می‌دهند. تفاوت اصلی شبکه بات با سایر انواع بدافزارها، وجود زیرساخت فرمان و کنترل است که به بات‌ها اجازه می‌دهد دستورات را از مدیر بات دریافت کنند. مدیر بات باید مطمئن شود که ساختار فرمان و کنترل به‌اندازه کافی قوی است که تعداد زیادی بات توزیع‌شده در سراسر جهان را مدیریت کند و همچنین در برابر هرگونه تلاش برای از بین بردن شبکه‌های بات مقاومت کند. از شبکه‌های بات در جرائم سازمان‌یافته به‌منظور نفوذ در دستگاه‌های امنیتی دولت‌ها، بانک‌ها و شرکت‌ها استفاده فراوانی می‌شود و کاربرد حملات انکار سرویس[۳] در آن بسیار متداول است. در سال‌های اخیر تحقیقات زیادی برای تشخیص و پیشگیری از شبکه‌های بات انجام‌شده است، اما شبکه‌های بات نیز به همان اندازه رشد و گسترش‌یافته‌اند.

*رایانامه نویسنده مسئول: Parsa@iust.ac.ir

[1] Botmaster
[2] Command and Control (C&C)
[3] Distributed Denial of Service (DDoS)

ناشر: دانشگاه جامع امام حسین (ع)

مهاجمین با استفاده از روش تشخیص شبکه بات مبتنی بر الگوریتم تولید دامنه از قرار گرفتن نام دامنه سرویس‌دهنده‌های فرمان و کنترل خود در فهرست‌های سیاه جلوگیری می‌کنند. بسیاری از روش‌های تشخیص شبکه بات، مبتنی بر تحلیل فعالیت گروهی شبکه‌های بات هستند [۲ و۳]، محققانی که با روش‌های متفاوتی دامنه‌های بدخواه را شناسایی کرده‌اند به این نتیجه رسیده‌اند که اغلب شبکه‌های بات مبتنی بر الگوریتم‌های تولید دامنه، فرضیه‌های مشابهی را ارائه می‌دهند ازجمله (الف) دامنه‌های تولیدشده توسط الگوریتم تولید دامنه، ساختار متفاوتی با دامنه‌های عادی دارند و (ب) الگوهای ترافیک سرویس‌دهنده نام دامنه شبکه‌های بات متفاوت با میزبان‌های قانونی است.



شکل (۱): شمای کلی یک شبکه بات

تحقیقات زیادی برای تشخیص الگوریتم‌های تولید دامنه و دسته‌بندی آن‌ها با به‌کارگیری صفات متعددی انجام‌شده است. مک‌گراس و گوپتا [۴] از ویژگی‌هایی مانند سوابق ویز[1]، ویژگی‌های لغوی[2] و آدرس‌های آی‌پی بدخواه شناخته‌شده استفاده کرده‌اند. محققان دیگر با به‌کارگیری درخت تصمیم‌گیری J48 [۵] و استفاده از ویژگی‌های مبتنی بر زمان، مبتنی بر پاسخ به سرویس‌دهنده نام دامنه[3] و مبتنی بر دامنه سعی در شناسایی الگوریتم‌های تولید دامنه داشته‌اند. ویژگی‌هایی مانند طول دامنه و نام میزبان برای شناسایی هرزنامه‌های تبلیغاتی مورداستفاده قرارگرفته است [۶]. آنتوناکاکیس و همکاران [۷] از ویژگی‌هایی مانند روش دو گرام[4]، توزیع خصوصیات[5] و ویژگی‌های ساختاری دامنه‌ها مانند طول نام دامنه و کلمات موجود در دامنه و فن‌هایی مانند رگرسیون[6] [۸]، درخت تصمیم‌گیری متناوب[7] [۹]، ماشین بردار پشتیبان[8] [۱۰] و شبکه‌های عصبی اسپایک تکامل‌یافته[9] استفاده کرده‌اند. علاوه بر این، روش‌های یادگیری عمیق و روش‌های استفاده‌کننده از توالی منابع محلی جهانی[10] با به‌کارگیری ویژگی‌های مبتنی بر دامنه و ویژگی‌های مبتنی بر آدرس اینترنتی برای تشخیص الگوریتم‌های تولید دامنه مورداستفاده قرارگرفته است [۱۱ و ۱۲].

ساختار این مقاله بدین شرح است: در بخش ۲، پژوهش‌های مرتبط با تشخیص دامنه‌های بدخواه و ناسالم تولیدشده توسط الگوریتم‌های تولید دامنه ارائه‌شده است. روش پیشنهادی شامل ارائه دو سناریوی استخراج ویژگی دستی و خودکار از مجموعه داده موجود، شبکه عصبی خود رمزگذار و ساختار شبکه خود رمزگذار عمیق پیشنهادی است که در بخش ۳ موردبحث قرارگرفته است. نتایج و اعتبارسنجی آزمایش‌های مختلف در بخش ۴ بررسی‌شده است. بخش ۵ به نتیجه‌گیری و کارهای آینده اختصاص‌یافته است.

۲- پژوهش‌های مرتبط

تحقیقات متعددی در حوزه شناسایی دامنه‌های بدخواه و مخرب از دامنه‌های سالم در شبکه‌های بات انجام‌شده است که هریک دارای مزایا و معایبی بوده‌اند و هرکدام از آن‌ها با دقت، صحت و سرعت‌های تشخیص متفاوتی این دامنه‌ها را شناسایی کرده‌اند که برخی از این تحقیقات در زیر اشاره‌شده است.

بیلگه و همکاران [۵]، روش اکسپوشر[11] را برای پیدا کردن دامنه‌های بدخواه معرفی کردند. به‌طور خاص، آن‌ها ابتدا ۱۵ ویژگی از یک دامنه و تمام آی‌پی‌هایی که دامنه به آن نگاشت شده است را استخراج کردند. سپس، آن‌ها با استفاده از درخت تصمیم‌گیری، هر دامنه داده‌شده را به‌عنوان دامنه بدخواه یا دامنه قانونی رده‌بندی کردند. آنتوناکاکیس و همکاران، سیستم پویایی را برای سرویس‌دهنده نام دامنه به نام نوتوس[12] معرفی کردند [۷]. نوتوس مدل‌هایی از دامنه‌های قانونی و بدخواه شناخته‌شده را با استفاده از ۱۸ ویژگی شبکه، ۱۷ ویژگی مبتنی بر دامنه و ۶ ویژگی مبتنی بر مشاهدات ایجاد می‌کند. اطلاعات قانونی شامل اطلاعات جمع‌آوری‌شده از چند سرویس‌دهنده نام دامنه بوده و اطلاعات بدخواه شامل دامنه‌های بدخواه و آی‌پی‌های به‌دست‌آمده از منابع مختلف بود. سپس، از این مدل‌ها برای محاسبه امتیاز یک دامنه جدید استفاده کرده‌اند.

آنتوناکاکیس و همکاران همچنین با استفاده از الگوریتم‌های دسته‌بندی و NXDomains جمع‌آوری‌شده از سرویس‌دهنده‌های محلی نام دامنه برای شناسایی بات‌های مبتنی بر الگوریتم‌های تولید دامنه استفاده کردند. الگوریتم دسته‌بندی برای یافتن دسته‌های بزرگ NXDomains استفاده‌شده که (الف) دارای

[1] Whois records
[2] Lexical characteristics
[3] Domain Name Server (DNS)
[4] Bigram
[5] Distribution of characters
[6] Regression
[7] Alternating Decision Tree
[8] Support Vector Machines (SVM)
[9] evolving Spiking Neural Networks (eSNNs)
[10] Universal Resource Locator (URLs)
[11] Exposure
[12] Notos

صفات زبانی مشابه هستند و (ب) توسط بات‌های متعدد در طول یک بازه زمانی مورد سوءاستفاده قرارگرفته‌اند. الگوریتم دسته‌بندی برای اختصاص دسته‌های تولیدشده به مدل‌های شناخته‌شده و تولیدشده توسط الگوریتم به‌کاررفته است.

کورتین و همکاران [۹] در مقاله خود یک اندازه‌گیری از پیچیدگی برای تشخیص دامنه‌های تولیدشده توسط الگوریتم‌های تولید دامنه به نام امتیاز اسمشورد[1] را ارائه دادند که نشان‌دهنده میزان شباهت دامنه‌های تولیدشده توسط الگوریتم نسبت به کلمات انگلیسی است. ازآنجاکه الگوریتم‌های تولید دامنه با امتیاز کلمات بالاتر به‌طورمعمول مشکل بیشتری برای تشخیص دامنه‌های بدخواه ایجاد می‌کنند، آن‌ها از یک مدل یادگیری ماشین ترکیبی متشکل از شبکه‌های عصبی مکرر[2] با استفاده از آزمون نرخ شباهت عمومی[3] ایجاد کرده و این مدل‌ها را با یک مدل رگرسیون منطقی تقویت کردند. مدل ترکیبی آن‌ها نسبت به روش‌های تشخیص الگوریتم‌های تولید دامنه موجود با امتیاز کلمات کلیدی بالاتری اقدام به تشخیص دامنه‌های بدخواه می‌کرد.

شیاوونی و همکاران [۱۳]، فونیکس[4] را برای تشخیص دامنه‌های سالم و عادی از دامنه‌های ناسالم و تولیدشده توسط الگوریتم با استفاده از ویژگی‌های زبانی و آی‌پی معرفی کردند. اولاً، فینیکس دامنه‌های قابل‌درکی را تولید می‌کند که توسط انسان تولید می‌شود و فرض می‌کند که دامنه تولیدشده توسط الگوریتم؛ این مدل‌ها را نقض می‌کند. بر اساس این مدل‌ها، دامنه تولیدشده توسط الگوریتم از فهرست سیاه شناخته‌شده استخراج می‌شود. پس‌ازآن، فونیکس این دامنه‌های استخراج‌شده را با استفاده از روابط دامنه به آی‌پی دسته‌بندی می‌کند.

در [۸] و [۱۴]، یادآو و همکاران، الگوریتم‌هایی را برای شناسایی دامنه‌های تولیدشده توسط الگوریتم‌های تولید دامنه معرفی کردند. فرضیه آن‌ها برای جلوگیری از تضاد با دامنه‌های موجود این بود که شبکه‌های بات فعلی از کلمات زبان قابل تلفظ استفاده نمی‌کنند. در مرحله بعدی، نویسندگان از سه روش برای دسته‌بندی پرس‌وجوهای سرویس‌دهنده نام دامنه استفاده کردند، به‌عنوان‌مثال، تمام دامنه‌هایی که با یک آدرس آی‌پی نشان داده می‌شوند باهم دسته‌بندی می‌شوند. آن‌ها محاسبات متعددی را برای مشخص کردن توزیع کاراکترهای الفبایی در هر دسته، انجام داده‌اند. این معیارها شامل آنتروپی اطلاعات[5]، شاخص جاکارت[6] و ویرایش فاصله[7] بود. درنهایت، معیارهای مذکور برای تمایز نام دامنه قانونی از دامنه‌های بدخواه استفاده شد. این روش تنها قادر به شناسایی دامنه‌های تولیدشده توسط الگوریتم نامشخص بود.

شریف‌نیای و آبادی [۳] روشی برای تشخیص شبکه‌های بات نسل جدید پیشنهاد دادند که از ترافیک سیستم نام دامنه برای محاسبه شهرت منفی میزبان‌های مشکوک استفاده می‌کند. در روش پیشنهادی آن‌ها، ابتدا در پایان هر پنجره زمانی، پرس‌وجوهای سیستم نام دامنه با ویژگی‌های مشابه انتخاب می‌شود. سپس میزبان‌های تولیدکننده نام دامنه با استفاده از الگوریتم تولید دامنه با توجه به توزیع کاراکترهای حرف و عددی در این پرس‌وجوها شناسایی‌شده و به ماتریس فعالیت‌های گروهی مشکوک اضافه می‌شوند. همچنین، اگر میزبان‌هایی که تعداد شکست‌ها در پرس‌وجوهای سیستم نام دامنه آن‌ها از یک آستانه مشخص گذر کند به ماتریس شکست‌های مشکوک اضافه می‌شوند. شهرت منفی میزبان‌ها در این دو ماتریس محاسبه‌شده و میزبان‌های دارای شهرت منفی بالا به‌عنوان میزبان‌های آلوده به بات تشخیص داده می‌شوند.

در سال ۲۰۲۰، رن و همکاران [۱۵]، چارچوب یادگیری عمیق[8] متشکل از شبکه عصبی حافظه کوتاه‌مدت ماندگار دوطرفه[9] و شبکه عصبی پیچشی[10] را برای شناسایی دامنه‌های تولیدشده توسط الگوریتم‌های تولید دامنه پیشنهاد کردند. در مرحله ابتدایی، شبکه عصبی پیچشی و شبکه عصبی حافظه کوتاه‌مدت ماندگار دوطرفه برای استخراج ویژگی‌های اطلاعات توالی دامنه استفاده‌شده و در مرحله بعدی، لایه توجه[11] برای اختصاص وزن مربوط به اطلاعات عمیق استخراج‌شده از نام دامنه استفاده‌شده است. درنهایت، وزن‌های مختلف ویژگی‌های نام دامنه برای انجام وظایف تشخیص و طبقه‌بندی در لایه خروجی قرار داده شد. نتایج، اثربخشی مدل آن‌ها در تشخیص دامنه‌های بدخواه را نشان داده است که به‌طور میانگین نرخ دقت و نرخ تشخیص ۸۳٪ گزارش‌شده است.

با توجه به مشکلات موجود در روش‌های مطرح‌شده ازجمله مدت‌زمان زیاد در تشخیص دامنه‌های تولیدشده توسط الگوریتم‌های تولید دامنه و نیز نرخ صحت، نرخ دقت و نرخ تشخیص پایین و نرخ خطای بالا در تشخیص دامنه‌های بدخواه و ناسالم، روشی جدید برای تشخیص دامنه‌های بدخواه و ناسالم از دامنه‌های سالم با نرخ صحت و دقت و نرخ تشخیص بالاتر در مدت‌زمان کوتاه‌تری ارائه می‌شود.

[1] Smashword
[2] Recurrent Neural Networks (RNNs)
[3] Generalized Likelihood Ratio Test (GLRT)
[4] Phoenix
[5] Information Entropy
[6] Jaccard Index
[7] Edit-distance
[8] Deep Learning Framework
[9] Bidirectional Long Short Term Memory (BiLSTM)
[10] Convolutional Neural Network (CNN)
[11] Attention Layer

۳- روش پیشنهادی

اخیراً در شبکه‌های بات از فهرست نام‌های دامنه سرویس‌دهندگان فرمان و کنترل به‌صورت پویا استفاده می‌شود. این فهرست پویا که توسط یک الگوریتم تولید دامنه ایجاد می‌شود به مهاجمان کمک می‌کند تا مکان سرویس‌دهندگان فرمان و کنترل را به‌صورت دوره‌ای تغییر داده و از قرار گرفتن آدرس‌های آن‌ها در فهرست‌های سیاه جلوگیری کند. با توجه به اینکه شناسایی دامنه‌های که هیچ‌گونه اطلاعی از الگوریتم تولید آن‌ها وجود ندارد بسیار دشوار است در این مقاله پیشنهادشده است ویژگی‌هایی از دامنه‌های موجود، استخراج‌شده و موردبررسی قرار گیرد تا بتوان در صورت تأثیرگذار بودن این ویژگی‌ها در افزایش میزان دقت تشخیص دامنه‌های بدخواه و تولیدشده توسط الگوریتم‌های تولید دامنه، در سامانه‌های تشخیص دامنه‌های بدخواه و ناسالم از این ویژگی‌ها و روش‌های استخراج پیشنهادی استفاده کرد. در بخش‌های بعدی، الگوریتم‌های تولید دامنه و دو سناریوی پیشنهادی استخراج ویژگی تأثیرگذار و شبکه عصبی عمیق خود رمزگذار موردبررسی قرار می‌گیرد.

۳-۱- الگوریتم‌های تولید دامنه

تمرکز الگوریتم‌های تولید دامنه برای ایجاد یک‌رشته تصادفی حاوی کاراکترها و اعداد به‌صورت ایستا است. برای تولید یک نام دامنه مناسب، دامنه سطح بالا[1] همچون.ir به رشته تصادفی تولیدشده افزوده می‌شود. آدرس‌های آی‌پی در فهرست سیاه قرارگرفته و مسدود می‌شوند. شبکه‌های بات مدرن از الگوریتم‌های تولید دامنه برای تولید یک زیرساخت فرمان و کنترل مقاوم استفاده کرده [۱۳] و شناسایی شبکه‌های بات تولیدشده با استفاده از دامنه‌های پویا برای شرکت‌های امنیتی اینترنتی دشوار است [۱۶].

با توجه به شناسایی دامنه‌های شبکه‌های بات در زمان اندک، دامنه‌های بکار گرفته‌شده در شبکه‌های بات دارای مدت حیات کوتاهی بوده و با سرعت تغییر می‌یابند؛ بنابراین استفاده از فهرست سیاه برای تشخیص این شبکه‌های بات مؤثر نیست. تمام شبکه‌های بات و سرویس‌دهنده‌های فرمان و کنترل که دارای زیرساخت یکسانی هستند، از یک الگوریتم یکسان استفاده کرده و همه آن‌ها یک دامنه یکسانی را به‌صورت مجزا تولید می‌کنند. زیرمجموعه‌ای از این دامنه‌ها توسط سرویس‌دهنده‌های فرمان و کنترل ذخیره‌شده و مدیر بات جهت ارتباط با شبکه‌های بات متعلق به خود، این دامنه‌ها را به‌کار می‌گیرد [۲]. برای پیچیده ساختن فرآیند شناسایی الگوریتم تولید دامنه، سرویس‌دهنده‌های فرمان و کنترل مرتباً دامنه‌های جدیدی را با استفاده از الگوریتم تولید دامنه تولید می‌کنند که این کار موجب تولید فهرست سیاه بزرگی شده و تشخیص آن‌ها سخت‌تر می‌شود [۳].

در این مقاله با توجه به استفاده از یادگیری عمیق و الگوریتم‌های یادگیری ماشین و عملکرد خوب آن‌ها در پیش‌بینی و تشخیص، باید ابتدا داده‌های موجود که نام دامنه‌ها هستند به داده‌هایی برای پذیرش توسط این الگوریتم‌ها تبدیل شوند. با توجه به ویژگی‌های مختلف داده‌ها، داده‌های مجموعه داده موجود با استفاده از دو سناریوی پیشنهادی به‌صورت دستی و خودکار پیش‌پردازش و آماده‌سازی شده و به‌عنوان ورودی شبکه عصبی خود رمزگذار و الگوریتم‌های یادگیری استفاده می‌شوند.

۳-۲- سناریوی استخراج ویژگی دستی پیشنهادی

در روش دستی استخراج داده‌ها، داده‌های مجموعه داده (نام دامنه‌ها) به یک جدول ساخت‌یافته تبدیل‌شده و به‌عنوان ورودی الگوریتم‌ها مورداستفاده قرار می‌گیرند. مهندسی ویژگی‌ها بر روی نام دامنه‌های مجموعه داده انجام‌یافته و درنهایت سه نوع کلی ویژگی‌ها به نام ویژگی‌های ساختاری، زبانی و آماری برای تولید مجموعه داده جدید به‌کار گرفته می‌شوند. در شبکه‌های باتی که از الگوریتم‌های تولید دامنه برای تولید نام‌های دامنه شبه تصادفی به‌منظور ارتباط با سرویس‌دهنده فرمان و کنترل استفاده می کنند با محاسبه آنتروپی شانون [۱۷]، امتیاز آنتروپی یک دامنه محاسبه‌شده و دامنه‌های ناسالم از دامنه‌های سالم تشخیص داده می‌شوند. میزان آنتروپی برای یک زیر دامنه، پس از محاسبه احتمال رخداد یک کاراکتر $P(x_i)$ دریک زیر دامنه و با استفاده از رابطه (۱) برای تمامی دامنه‌ها به‌دست می‌آید:

$$Entropy = -\sum_{i=1}^{n}\left(P(x_i)\log_2 P(x_i)\right) \qquad (۱)$$

در این مقاله از مقدار آنتروپی محاسبه‌شده برای هر دامنه به‌عنوان یک ویژگی آماری از مجموعه ویژگی‌ها جهت شناسایی دامنه‌های بدخواه استفاده‌شده است. جدول (۱) ویژگی‌های زبانی، جدول (۲) ویژگی‌های ساختاری و جدول (۳) ویژگی‌های آماری استخراج‌شده از مجموعه داده موجود را به همراه دو مثال نشان می‌دهد.

[1] Top Domain Level (TLD)

جدول (۱): ویژگی‌های زبانی[1]

ویژگی	توضیح	نوع ویژگی	مثال ۱ codexsprawl.wordpress.com	مثال ۲ gxy7p2uyuxhtp3oudp.ru
Contains_digit	دارای رقم در زیر دامنه	دودویی	۰	۱
Vowel_ratio	نرخ حروف صدادار در زیر دامنه (A, E, I, O, U)	عدد اعشاری	۰/۲۵	۰/۲۶۶
Digit_ratio	نرخ ارقام در زیر دامنه	عدد اعشاری	۰/۰	۰/۱۶۶

جدول (۲): ویژگی‌های ساختاری[2]

ویژگی	توضیح	نوع ویژگی	مثال ۱ codexsprawl.wordpress.com	مثال ۲ gxy7p2uyuxhtp3oudp.ru
HwP	دارای پیشوند www	دودویی	۰	۰
DNL	طول نام دامنه	عدد صحیح	۲۵	۲۱
SLM	میانگین طول زیر دامنه	عدد اعشاری	۱۰/۰	۱۸/۰
NoS	تعداد زیر دامنه‌ها	عدد صحیح	۲	۱
CTS	دارای TLD به‌عنوان زیر دامنه	دودویی	۰	۰
CSCS	دارای زیر دامنه تک کاراکتری	دودویی	۰	۰
UR	نسبت زیر دامنه به‌کل دامنه	عدد اعشاری	۰/۰	۰/۰
CIPA	دارای آدرس آی پی	دودویی	۰	۰
HVTLD	دارای TLD معتبر	دودویی	۱	۱

جدول (۳): ویژگی‌های آماری[3]

ویژگی	توضیح	نوع ویژگی	مثال ۱ codexsprawl.wordpress.com	مثال ۲ gxy7p2uyuxhtp3oudp.ru
RRC	نرخ تعداد تکرار کاراکترها در زیر دامنه	عدد اعشاری	۰/۶۳۶	۰/۳۳۳
RCC	نرخ حروف بی‌صدا در زیر دامنه	عدد اعشاری	۰/۳۸۸	۰/۴۷۶
RCD	نرخ ارقام متوالی در زیر دامنه	عدد اعشاری	۰/۰	۰/۰
Entropy	آنتروپی زیر دامنه	عدد اعشاری	۳/۴۶۳	۳/۴۱۹

همان‌گونه که در جدول (۱) نشان داده شده است، ویژگی Contains_digit وجود یا عدم وجود رقم را در دامنه مشخص می‌کند که در مثال ۱ با توجه به عدم وجود رقم این ویژگی مقدار ۰ و در مثال ۲ با توجه به وجود رقم مقدار ویژگی برابر با ۱ است. ویژگی Vowel_ratio، نرخ حروف صدادار در زیر دامنه (بخش بدون دامنه سطح بالا یا TLD مانندcom. وru.) را تعیین می‌کند که از تقسیم تعداد حروف صدادار به تعداد کل حروف زیر دامنه حاصل می‌شود و مقدار آن در مثال ۱ برابر با ۰/۲۵ (تعداد حروف صدادار برابر با ۳ و تعداد کل حروف زیر دامنه برابر با ۱۸) و در مثال ۲ برابر با ۰/۲۶۶ است. ویژگی Digit_ratio، نرخ ارقام موجود در زیر دامنه را تعیین می‌کند که حاصل تقسیم تعداد ارقام به تعداد کل حروف زیر دامنه است و مقدار آن در مثال ۱ برابر با ۰,۰ (تعداد ارقام زیر دامنه برابر با ۰ و تعداد کل حروف زیر دامنه برابر با ۱۸) و در مثال ۲ برابر با ۰/۱۶۶ است.

در جدول (۲)، ویژگی HwP، وجود یا عدم وجود پیشوند www را در دامنه تعیین می‌کند که در صورت وجود پیشوند مقدار ۱ و در صورت عدم وجود پیشوند مقدار ۰ دارد. ویژگی DNL، تعداد کل کاراکترهای دامنه اعم از حروف، ارقام و نقطه‌ها را تعیین می‌کند. ویژگی SLM، میانگین طول کاراکترهای زیر دامنه را مشخص می‌کند برای مثال ۱ این مقدار برابر ۹ است و از حاصل جمع طول زیر دامنه codexsprawl (با طول ۱۱) و طول

[1] Linguistic
[2] Structural
[3] Statistical

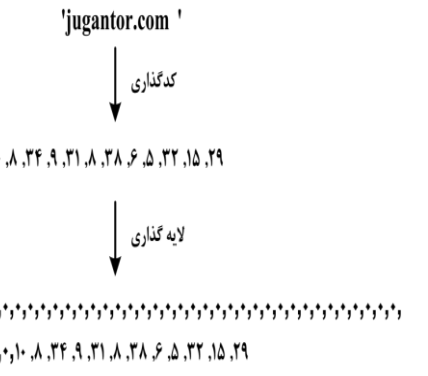
زیر دامنه wordpress (با طول ۹) تقسیم‌بر تعداد زیر دامنه‌ها (NoS) که در این مثال برابر با ۲ است، به‌دست می‌آید. ویژگی CTS، برای تعیین وجود پسوند و دامنه سطح بالا به‌عنوان زیر دامنه در دامنه اصلی مورداستفاده قرار می‌گیرد. برای مثال در دامنه Azad**.com**.com، با توجه به وجود دامنه سطح بالای com. به‌عنوان زیر دامنه در دامنه اصلی مقدار ویژگی CTS برابر با ۱ است. درصورتی‌که زیر دامنه تک کاراکتری در دامنه وجود داشته باشد مقدار ویژگی CSCS، برابر با یک و در غیر این صورت برابر با ۰ خواهد بود. مقدار ویژگی CIPA، در صورتی برابر با ۱ خواهد بود که دامنه حاوی آدرس آی‌پی باشد. ویژگی HVTLD، بیانگر وجود دامنه سطح بالا (TLD) در دامنه است.

در جدول (۳)، ویژگی RRC، نرخ تعداد تکرار کاراکترها در زیر دامنه و ویژگی RCC، نرخ حروف بی‌صدا (تمامی حروف الفبای انگلیسی به‌جز حروف صدادار) در زیر دامنه را تعیین می‌کند. همچنین ویژگی RCD، نرخ ارقام متوالی در زیر دامنه (برای مثال ارقام متوالی 12 در زیر دامنه Azad12iau) را نشان داده و ویژگی Entropy، آنتروپی زیر دامنه محاسبه‌شده با استفاده از رابطه (۱) را ارائه می‌دهد.

۳-۳- سناریوی استخراج ویژگی خودکار پیشنهادی

روش استخراج ویژگی‌های خودکار از ساخت چندین شاخص برای نگاشت ویژگی‌ها به شاخص‌ها استفاده می‌کند. بر اساس این ایده، می‌توان به‌طور مستقیم از شاخص‌ها به‌عنوان یک دنباله[1] استفاده کرده و یا آن‌ها را با روش کدگذاری به کد تبدیل کرد. داده‌های دامنه موجود در این مقاله به شکل زبان طبیعی نیستند که به‌راحتی بتوان آن‌ها را نشانه‌گذاری کرد. همچنین، هیچ روشی برای ایجاد دامنه‌ای که به پردازش بهتر کمک کند وجود ندارد. دامنه‌های فرعی شامل منطق متنی نیستند. یک دامنه به تولیدشده توسط الگوریتم تولید دامنه معمولاً متشکل از ترکیبات تصادفی از کاراکترها و اعداد است و هیچ ارتباطی بین ارقام کنار هم وجود ندارد؛ بنابراین، هرگز نمی‌توان یک معیار نشانه‌گذاری برای دامنه‌های تولیدشده توسط الگوریتم‌های تولید دامنه یافت. واژه‌نامه‌ای[2] از تمام حروف، اعداد و کاراکترهای مجاز را که در دامنه‌ها استفاده می‌شود تولید کرده و برای هر کاراکتر عددی و یا غیر عددی، مقداری اختصاص داده می‌شود. واژه‌نامه همانند یک شاخص است و می‌توان هر یک از کاراکترهای دامنه موجود در مجموعه داده را به شاخص‌هایی که مطابق با واژه‌نامه هستند، تبدیل کرده و داده‌ها به بردارهایی با طول n تبدیل می‌شوند.

ازآنجاکه شکل ورودی شبکه‌های عصبی باید یکنواخت باشد، باید تمام نمونه‌ها (دامنه‌ها) دارای طول یکسان با n عضو باشند؛ بنابراین در این مقاله از روش لایه گذاری[3] برای افزودن مقادیر اضافی به بردارهای نمونه برای یکسان‌سازی طول داده‌های ورودی استفاده می‌شود، (شکل (۲)). بر اساس مجموعه داده موجود، با در نظر گرفتن طول دامنه‌های موجود، طول لایه گذاری n=۷۳ در نظر گرفته می‌شود زیرا حداکثر طول دامنه موجود در مجموعه داده برابر با ۷۳ است.



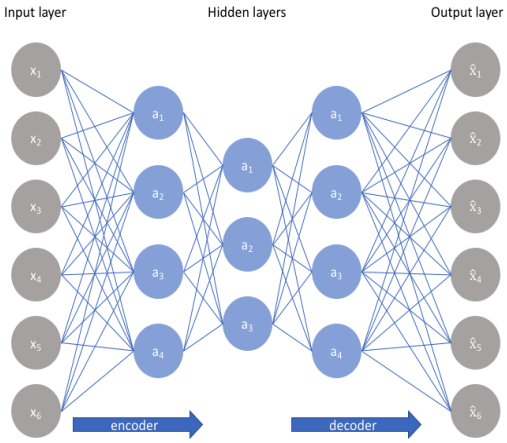
شکل (۲): مراحل انجام لایه گذاری پیشنهادی

۴-۳- ساختار شبکه‌های عصبی خود رمزگذار پیشنهادی

شبکه‌های عصبی خود رمزگذار[4] [۱۸]، شبکه‌های ساده یادگیری هستند که باهدف تبدیل ورودی به خروجی بدون کمترین تغییر پیاده‌سازی می‌شوند. در عین سادگی شبکه‌های خود رمزگذار نقش به سزایی را در یادگیری ماشین ایفا می‌کنند. به همراه قوانین یادگیری هب[5]، شبکه‌های خود رمزگذار، الگوی اصلی یادگیری بدون نظارت را تشکیل می‌دهند. شبکه‌های خود رمزگذار در معماری عمیق به فرم ماشین محدود بولتزمن[6] موردتوجه قرارگرفته‌اند. در این دسته مسائل، برچسبی برای توصیف داده‌ها وجود ندارد (برخلاف یادگیری با نظارت که در آن‌ها از برچسب‌هایی برای توصیف داده‌ها استفاده می‌شود). شبکه خود رمزگذار یک شبکه عصبی است که مجموعه‌ای از داده‌های بدون برچسب را دریافت می‌کند و با رمز کردن[7] آن‌ها سعی در بازنمایی مجدد ورودی‌ها در خروجی می‌کند به صورتی که کمترین اختلاف ممکن را با مقدار ورودی داشته باشند. شکل (۳) یک شبکه خود رمزگذار را نشان می‌دهد. شبکه در جهتی آموزش می‌بیند که وزن‌های تولیدشده در لایه‌ها، سبب شود خروجی با ورودی حداقل اختلاف ممکن را داشته باشد و در حالت ایده‌آل ممکن است برابر شوند.

[1] Sequence
[2] Dictionary
[3] Padding
[4] Autoencoder
[5] Hebbian learning
[6] Boltzman Machine
[7] Encoding

ساختار شبکه خود رمزگذار به دو بخش رمزگذاری و رمزگشایی تقسیم می‌شود. در بخش رمزگذاری، داده‌های ورودی به فضای ویژگی‌ها نگاشت می‌شوند و در بخش رمزگشایی از فضای ویژگی مجدداً به حالت ابتدایی خود تبدیل می‌شوند. درواقع بخش اصلی یک خود رمزگذار، لایه‌ی پنهان میانی است که به‌عنوان ویژگی استخراج‌شده برای دسته‌بندی استفاده می‌شود.



شکل (۳): نمونه‌ای از یک شبکه خود رمزگذار

برای ساختار شبکه خود رمزگذار بکار گرفته‌شده از ۳ لایه رمزگذاری و ۳ لایه رمزگشایی استفاده‌شده است. برای پیاده‌سازی لایه‌های رمزگذاری و رمزگشایی از تابع فعال‌سازی تانژانت هیپربولیک[1] و برای لایه‌های میانی شبکه خود رمزگذار از تابع فعال‌سازی واحد خطی اصلاح‌شده[2] استفاده‌شده است. همچنین از بهینه‌ساز آدام[3] [۱۹] برای آموزش شبکه استفاده‌شده است. بر اساس تعریف ارائه‌شده در مرجع [۲۰]، خود رمزگذار تابعی است که عمدتاً از دو بخش تشکیل می‌شود:

- رمزگذار[4]: که یک تابع استخراج ویژگی بوده و بردار ویژگی را از روی ورودی‌ها محاسبه می‌کند. بنابراین اگر بردار ویژگی با h، تابع رمزگذار با f_θ و مجموعه داده‌ها با $x^{(i)}$ها نمایش داده شود، رابطه (۲) برقرار خواهد بود:

$$h^{(t)} = f_\theta(x^{(t)}), \qquad x^{(t)} = \{x^{(1)}, \ldots, x^{(T)}\} \qquad (۲)$$

به‌طوری‌که h بردار ویژگی یا کد محاسبه‌شده از x است.

- رمزگشا[5]: تابعی است که آن را با g_θ نمایش داده و با استفاده از رابطه (۳) نگاشتی از فضای ویژگی را به فضای ورودی انجام می‌دهد.

$$r = g_\theta(h) \qquad (۳)$$

درواقع از یک تابع احتمال مشخص، مدل‌های احتمالاتی تعریف‌شده و برای حداکثر کردن تشابه داده‌ها (اغلب به‌صورت تقریبی)، آموزش داده می‌شوند. شبکه‌های خود رمزگذار با استفاده از اصول آموزشی متفاوتی، آموزش داده می‌شوند. یادگیری مجموعه پارامترهای θ رمزگذار و رمزگشا به‌طور مشابه برای بازسازی ورودی اصلی انجام می‌شود، یعنی تلاش می‌شود تا خطای بازسازی L(x, r)، روی r حداقل شود. این کار با بازسازی x با بیش آموزش و اندازه‌گیری اختلاف میان x و r انجام می‌شود. به‌طور خلاصه آموزش شبکه خود رمزگذار شامل پیدا کردن بردار پارامتر θ برای حداقل‌سازی خطای بازسازی است که در رابطه (۴) ارائه‌شده است:

$$J_{AE}(\theta) = \sum_t L(x^{(t)},\ g_\theta(f_\theta(x^{(t)}))) \qquad (۴)$$

به حداقل رساندن این مقدار نیز معمولاً به روش گرادیان نزولی تصادفی مشابه روش آموزش پرسپترون چندلایه انجام می‌شود. معمولاً شبکه‌های خود رمزگذار برای کاهش بعد[6] و یا استخراج ویژگی مورداستفاده قرار می‌گیرند. در مرجع [۲۱]، ساختار جدیدی از شبکه‌های خود رمزگذار چندلایه مبتنی بر ساختار متقارن، نسبت به شبکه عصبی خود رمزگذار عادی، برای کاهش بعد استفاده‌شده است. این ساختار جدید تعداد وزن‌های لازم برای تنظیم را کاهش داده و درنتیجه توانسته است هزینه محاسباتی را نیز متعاقباً کاهش دهد.

- شبکه خود رمزگذار عمیق

در حال حاضر به‌راحتی می‌توان شبکه‌های بسیار عمیق را توسط پردازنده‌های گرافیکی آموزش داد برخی از این شبکه‌ها از جنس خود رمزگذارهای عمیق هستند. اگر شبکه عصبی سازنده یک خود رمزگذار، شبکه‌ای عمیق باشد آن را شبکه خود رمزگذار عمیق گویند. در این معماری تعداد لایه‌های پنهان شبکه بیش از یک‌لایه است.

با توجه به این‌که تعداد داده‌های ورودی در سناریوی دستی در استخراج ویژگی‌ها برابر با ۱۶ است تعداد گره‌های لایه ورودی و لایه خروجی ۱۶ در نظر گرفته‌شده است (شکل ۴).

همچنین به دلیل انجام عمل لایه‌گذاری و ثابت نگه‌داشتن تعداد ورودی‌های سناریوی خودکار در استخراج ویژگی‌ها، تعداد گره‌های لایه ورودی و لایه خروجی ۷۳ در نظر گرفته شده است (شکل ۵).

[1] Tangent hyperbolic (Tanh)
[2] Rectified Linear Unit (ReLU)
[3] Adam
[4] Encoder
[5] Decoder
[6] Dimension Reduction

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 16)	0
dense_1 (Dense)	(None, 12)	204
dense_2 (Dense)	(None, 8)	104
dense_3 (Dense)	(None, 4)	36
dense_4 (Dense)	(None, 8)	40
dense_5 (Dense)	(None, 12)	108
dense_6 (Dense)	(None, 16)	208

شکل (۴): ساختار شبکه خود رمزگذار عمیق برای سناریوی استخراج ویژگی دستی

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 73)	0
dense_1 (Dense)	(None, 55)	4070
dense_2 (Dense)	(None, 37)	2072
dense_3 (Dense)	(None, 19)	722
dense_4 (Dense)	(None, 37)	740
dense_5 (Dense)	(None, 55)	2090
dense_6 (Dense)	(None, 73)	4088

شکل (۵): ساختار شبکه خود رمزگذار عمیق برای سناریوی استخراج ویژگی خودکار

۴- نتایج و اعتبار سنجی

۴-۱- جزئیات پیاده‌سازی

برای انجام آزمایش‌های مختلف، رایانه با پردازنده شرکت اینتل با ۸ هسته Core i7-2670QM با سرعت ۲,۲۰ گیگاهرتز، حافظه اصلی ۸ گیگابایت و کارت گرافیکی NVIDIA GeForce با ۲ گیگابایت حافظه مورداستفاده قرارگرفته است. شبکه‌های عصبی خود رمزگذار با سناریوهای استخراج ویژگی دستی و خودکار با استفاده از زبان برنامه‌نویسی پایتون نسخه ۳,۶ در محیط ژوپیتر نوت‌بوک[1] و تنسورفلو[2] [۲۲] پیاده‌سازی و آموزش داده‌شده و از ابزار کراس نسخه ۲,۲,۲ [۲۳] که یک واسط برنامه‌نویسی کاربردی[3] برای نمونه سریع از تنسورفلو نسخه ۱,۱۰,۰ در پردازنده است، استفاده شده است. پلتفرم مورداستفاده برای یادگیری، ویندوز ۷ بوده و بسیاری از آزمایش‌ها برای تعیین پارامترهای تنظیم مناسب این الگوریتم‌ها انجام یافته است.

[1] Jupyter Notebook
[2] Tensorflow
[3] Application programming Interface (API)

۴-۲- مجموعه داده

در این مقاله از سه مجموعه داده اولیه برای تولید مجموعه داده جدید استفاده‌شده است که عبارت‌اند از:

۱. داده‌های سالم (دامنه‌های پاک) مجموعه آلکسا [۲۴]
۲. داده‌های بدخواه (دامنه‌های ناسالم):
الف. مجموعه داده بامبنک [۲۵]
ب. مجموعه داده آزمایشگاه ۳۶۰ [۲۶]

با ترکیب و درهم‌سازی سه مجموعه داده فوق، مجموعه داده جدیدی متشکل از دامنه‌های سالم و ناسالم برای استخراج ویژگی‌های دو سناریوی دستی و خودکار اشاره‌شده در بخش‌های ۳-۲ و ۳-۳ و به‌کارگیری در برخی از الگوریتم‌های یادگیری ماشین مشهور برای مقایسه نتایج به‌دست‌آمده، تولید شد. این مجموعه داده حاوی ۲,۴۵۸,۸۳۶ رکورد (دامنه) بوده و پس از به‌کارگیری دو سناریوی استخراج ویژگی دستی و خودکار، دو مجموعه داده جدید دیگری برای انجام آزمایش‌ها و ارزیابی عملکرد شبکه عصبی خود رمزگذار عمیق و الگوریتم‌های یادگیری ماشین با ۱۰۰ هزار رکورد تصادفی انتخاب شده از تمام رکوردهای مجموعه داده استفاده شد. برای انجام آزمایش‌ها، ۶۶ درصد از مجموعه داده برای آموزش شبکه عصبی خود رمزگذار عمیق و الگوریتم‌های یادگیری ماشین و ۳۳ درصد باقیمانده جهت آزمون تخصیص داده شد.

۴-۳. ارزیابی عملکرد و نتایج

برای ارزیابی شبکه عصبی خود رمزگذار عمیق با دو سناریوی پیشنهادی و برخی از الگوریتم‌های یادگیری ماشین نظارتی از روش اعتبار سنجی متقابل[4] با K-fold=۱۰ استفاده شد. با این روش میزان عملکرد الگوریتم‌ها به‌صورت دقیق‌تر مورد ارزیابی قرار گرفت. میانگین و انحراف معیار اجرای نتایج، با ۳۰ بار آزمایش بر روی مجموعه داده محاسبه‌شده و برای شبکه عصبی خود رمزگذار عمیق و الگوریتم‌های یادگیری ماشین ارائه‌شده است.

۴-۳-۱. سنجه‌های ارزیابی

یکی از مهم‌ترین مراحل پس از طراحی یا ساخت مدل، ارزیابی کارایی[5] آن است. در شبکه‌های عصبی و الگوریتم‌های یادگیری ماشین برای ارزیابی عملکرد هر یک از الگوریتم‌ها، سنجه‌هایی ازجمله نرخ صحت[6]، نرخ دقت[7]، نرخ تشخیص[8] یا نرخ بازخوانی[9]، میانگین هارمونی[10] و نرخ مثبت نادرست[11] استفاده می‌شود. پیش

[4] Cross Validation (C-V)
[5] Performance Evaluation
[6] Accuracy
[7] Precision
[8] True Positive Rate (TPR)
[9] Recall
[10] F-Measure
[11] False Positive Rate (FPR)

از محاسبه سنجه‌های ذکرشده باید تعاریف دیگری ارائه شود که عبارت‌اند از:

- **مثبت درست (TP):** تعداد دامنه‌های ناسالمی که به‌درستی به‌عنوان دامنه ناسالم شناسایی‌شده‌اند.
- **مثبت نادرست (FP):** تعداد دامنه‌های سالمی که به‌اشتباه به‌عنوان دامنه‌های ناسالم شناسایی‌شده‌اند.
- **منفی درست (TN):** تعداد دامنه‌های سالمی که به‌درستی به‌عنوان دامنه عادی و سالم شناسایی‌شده‌اند.
- **منفی نادرست (FN):** تعداد دامنه‌های ناسالمی که به‌اشتباه به‌عنوان دامنه‌های عادی و سالم شناسایی‌شده‌اند.

• **نرخ صحت:** درصد پیش‌بینی‌های درست تمام دامنه‌ها را نشان می‌دهد.

$$Accuracy\left(ACC\right)=\frac{TP+TN}{TP+TN+FP+FN} \quad (۵)$$

• **نرخ دقت:** نشان‌دهنده درصد دامنه‌هایی است که به‌درستی به‌عنوان دامنه ناسالم دسته‌بندی‌شده‌اند.

$$Precision\left(RPC\right)=\frac{TP}{TP+FP} \quad (۶)$$

• **نرخ تشخیص یا نرخ بازخوانی:** نشان‌دهنده درصد دامنه‌های ناسالمی است که به‌درستی به‌عنوان یک دامنه ناسالم پیش‌بینی‌شده است.

$$True\ Posetive\ Rate\left(TPR\right)=\frac{TP}{TP+FN} \quad (۷)$$

• **میانگین هارمونی:** این سنجه یک اندازه‌گیری از دقت آزمون بوده و از هر دو معیار بازخوانی و دقت آزمون برای محاسبه امتیاز استفاده می‌کند.

$$F-measure\left(F1\right)=\frac{2\times PRC\times RCL}{PRC+RCL} \quad (۸)$$

• **نرخ مثبت نادرست:** نشان‌دهنده درصد دامنه‌های سالمی است که به‌اشتباه به‌عنوان دامنه‌های ناسالم دسته‌بندی‌شده‌اند.

$$FPR=\frac{FP}{FP+TN} \quad (۹)$$

۴-۳-۲- نتایج

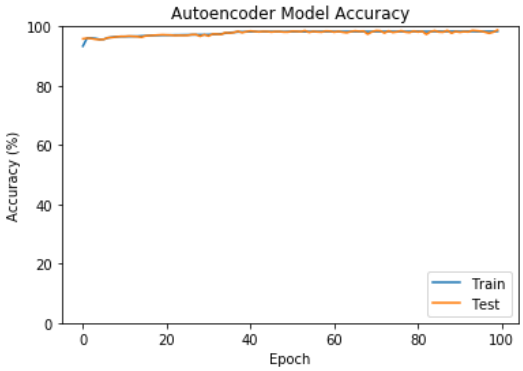
جدول (۴) و شکل‌های (۶) تا (۱۰)، نرخ صحت، نرخ دقت، نرخ تشخیص یا بازخوانی، نرخ میانگین هارمونی، نرخ مثبت نادرست و مدت‌زمان تشخیص توسط شبکه عصبی خود رمزگذار با دو سناریوی استخراج ویژگی دستی و استخراج ویژگی خودکار در مقایسه با برخی از الگوریتم‌های یادگیری ماشین نظارتی را نشان می‌دهد. با انجام آزمایش‌های متعدد نتایج ارائه‌شده در جدول (۴) حاصل‌شده است. این نتایج نشان می‌دهند شبکه عصبی خود رمزگذار عمیق در دو سناریوی استخراج دستی و خودکار ویژگی از مجموعه داده و همچنین الگوریتم جنگل تصادفی نسبت به سایر الگوریتم‌ها عملکرد بهتری ارائه می‌دهند. با تغییر پارامترهای تنظیم شبکه عصبی خود رمزگذار و الگوریتم‌های یادگیری ماشین نظارتی استفاده‌شده، نتایج متفاوتی توسط هر یک از این الگوریتم‌ها حاصل شد. با استفاده از شبکه عصبی خود رمزگذار عمیق در سناریوی استخراج خودکار ویژگی در مدت‌زمان قابل‌قبول ۲۰ ثانیه، نرخ صحت برابر با ۹۸,۶۱ درصد، نرخ تشخیص برابر با ۹۴/۲۷ درصد و میانگین هارمونی ۹۴/۱۴ درصد به‌دست‌آمده که درصد بالاتری نسبت به سایر الگوریتم‌ها ارائه می‌دهد و همچنین الگوریتم جنگل تصادفی نرخ دقت بالاتری برابر با ۹۴/۱۳ درصد نسبت به سایر الگوریتم‌ها ارائه می‌دهد که حاکی از دقت این الگوریتم است.

جدول (۴): مقایسه نتایج شبکه عصبی خود رمزگذار عمیق با دو سناریوی استخراج ویژگی دستی و خودکار و نتایج حاصل از الگوریتم‌های یادگیری ماشین

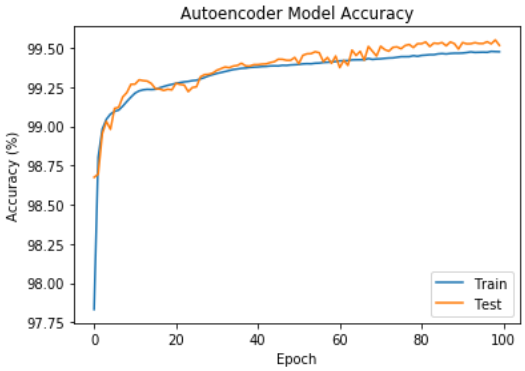
معیار / الگوریتم	جنگل تصادفی[1]	درخت تصمیم[2]	دسته‌بند بیز[3]	رگرسیون منطقی[4]	ماشین بردار پشتیبان[5]	K نزدیک‌ترین همسایه[6]	تقویتی تطبیقی[7]	سناریوی دستی	سناریوی خودکار
نرخ صحت	۸۹,۹۷ ± ۰,۱۱	۸۸,۱۴ ± ۰,۱	۷۹,۷۶ ± ۰,۰۵	۸۲,۱۲ ± ۰,۱۱	۸۷,۰۰ ± ۰,۱۱	۸۸,۰۷ ± ۰,۵۸	۸۳,۲۶ ± ۰,۰۶	۹۵,۱۵ ± ۰,۳۲	**۹۸,۶۱ ± ۰,۴۵**
نرخ دقت	**۹۴,۱۳ ± ۰,۱۲**	۹۳,۳۸ ± ۰,۲۵	۶۱,۶۲ ± ۰,۱۷	۸۷,۳۲ ± ۰,۰۲	۹۱,۶۸ ± ۰,۱۱	۹۲,۱۳ ± ۰,۲۷	۸۶,۸۴ ± ۰,۴	۹۲,۳۲ ± ۰,۸۳	۹۴,۰۲ ± ۲,۱۷
نرخ تشخیص	۹۱,۷۶ ± ۰,۱۷	۸۹,۱۱ ± ۰,۰۲	۷۹,۴۵ ± ۰,۱۵	۸۱,۱۶ ± ۰,۲۸	۸۹,۴۹ ± ۰,۰۱	۹۰,۰۸ ± ۰,۴۴	۸۹,۰۳ ± ۰,۱۴	۹۱,۲۹ ± ۰,۳۴	**۹۴,۲۷ ± ۱,۹۱**
میانگین هارمونی	۹۲,۹۱ ± ۰,۱۱	۹۱,۲۰ ± ۰,۱۲	۶۹,۴۱ ± ۰,۱۳	۸۴,۱۳ ± ۰,۱۵	۹۰,۵۷ ± ۰,۰۵	۹۱,۰۹ ± ۰,۲۵	۸۷,۹۲ ± ۰,۲۳	۹۱,۸۰ ± ۰,۴۶	**۹۴,۱۴ ± ۱,۹۵**
نرخ مثبت نادرست	۰,۳۵۲ ± ۰,۰۲۲	۰,۳۷۳ ± ۰,۰۰۶	۰,۴۱۴ ± ۰,۰۱۹	۰,۴۲۵ ± ۰,۰۱۱	۰,۳۶۲ ± ۰,۰۰۹	۰,۳۶۰ ± ۰,۰۸۸	۰,۳۹۳ ± ۰,۱۵۸	۰,۰۲۴ ± ۰,۰۲۹	**۰,۰۰۸ ± ۰,۰۰۳**
زمان تشخیص	۴۶,۷۴۴	۱۴,۵۹۵	۳,۷۵۸	۲۴,۳۸	۱۳۲,۷	۱۰۶,۶۵	۱۱۱,۷۸	**۲,۱۰۲**	۲۰,۰۱۷

* بهترین نتایج به‌صورت توپر مشخص‌شده‌اند.

[1] Random Forest (RF)
[2] Decision Tree (DT)
[3] Bayesian
[4] Logistic Regression (LR)
[5] Support Vector Machine (SVM)
[6] K-Nearest Neighbor
[7] Adaboost

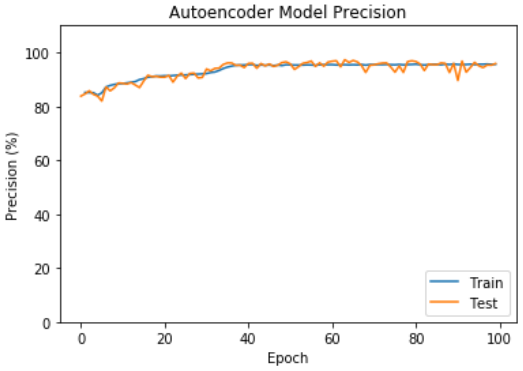


شبکه خود رمزگذار با انتخاب ویژگی به‌صورت دستی

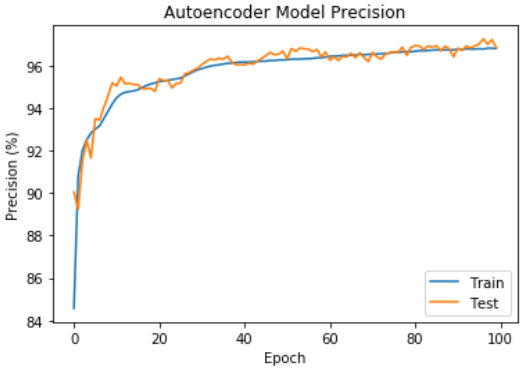


شبکه خود رمزگذار با انتخاب ویژگی به‌صورت خودکار

شکل (۶): نمودار مقایسه بر اساس معیار ارزیابی نرخ صحت

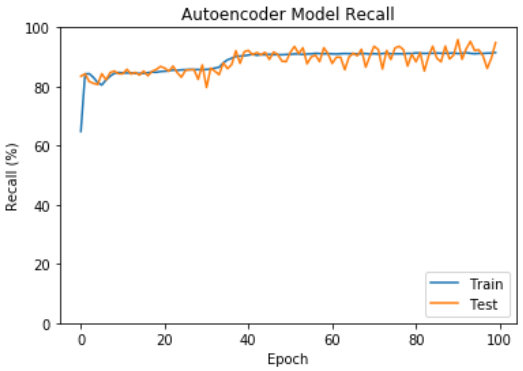


شبکه خود رمزگذار با انتخاب ویژگی به‌صورت دستی

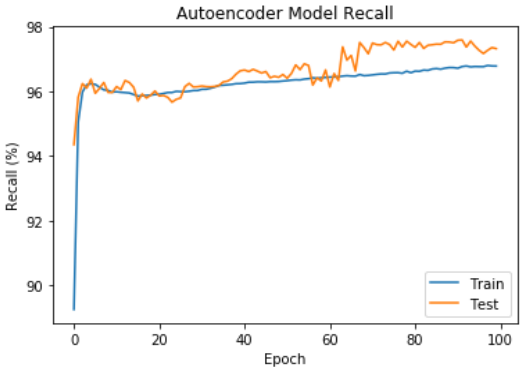


شبکه خود رمزگذار با انتخاب ویژگی به‌صورت خودکار

شکل (۷): نمودار مقایسه بر اساس معیار ارزیابی نرخ دقت

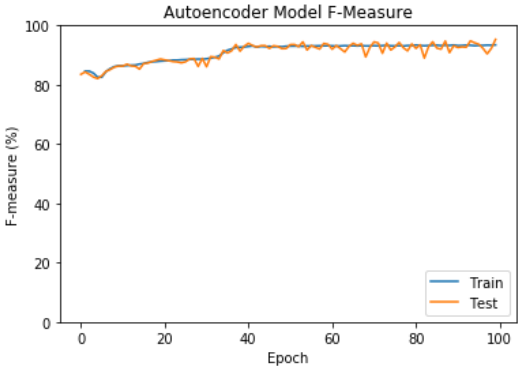


شبکه خود رمزگذار با انتخاب ویژگی به‌صورت دستی

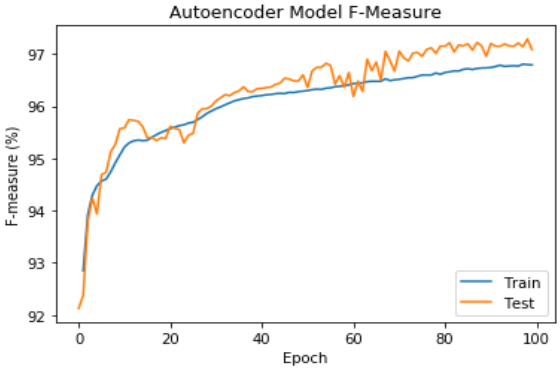


شبکه خود رمزگذار با انتخاب ویژگی به‌صورت خودکار

شکل (۸): نمودار مقایسه بر اساس معیار ارزیابی نرخ تشخیص یا بازخوانی

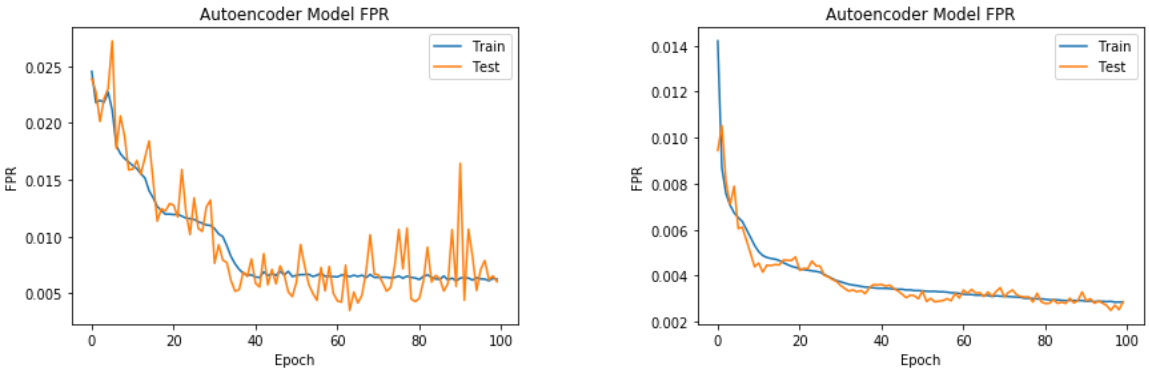


شبکه خود رمزگذار با انتخاب ویژگی به‌صورت دستی



شبکه خود رمزگذار با انتخاب ویژگی به‌صورت خودکار

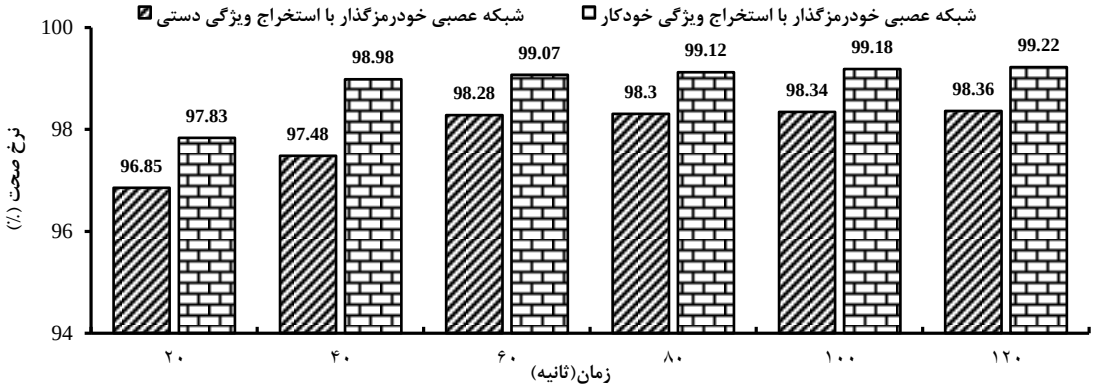
شکل (۹): نمودار مقایسه بر اساس معیار ارزیابی میانگین هارمونی



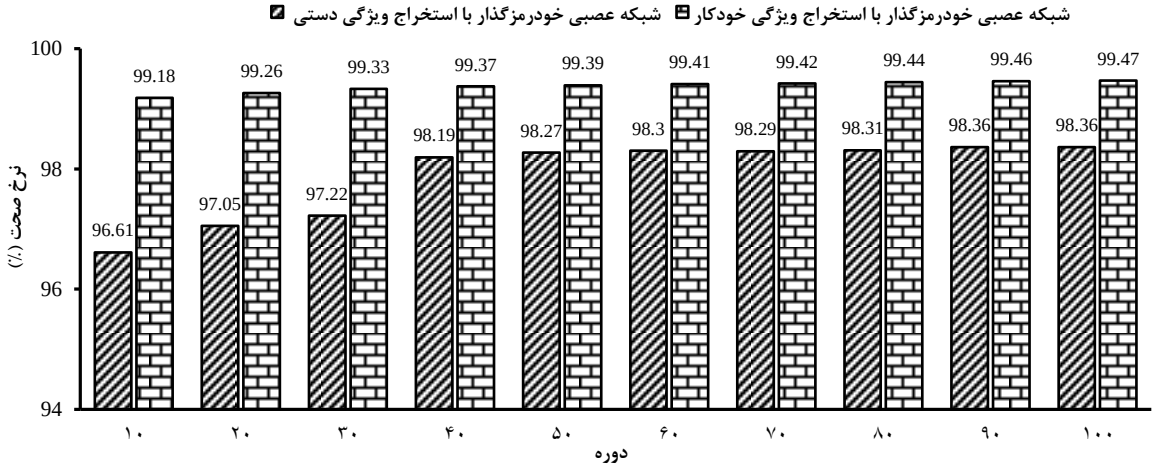
شبکه خود رمزگذار با انتخاب ویژگی به‌صورت خودکار — شبکه خود رمزگذار با انتخاب ویژگی به‌صورت دستی

شکل (۱۰): نمودار مقایسه بر اساس معیار ارزیابی نرخ مثبت نادرست

آزمایش‌های مختلفی برای بررسی تأثیر زمان تشخیص (برحسب ثانیه) و دوره اجرای (تعداد اجرا) شبکه عصبی خود رمزگذار عمیق بر نرخ صحت، انجام‌یافته است که در شکل‌های (۱۱ و ۱۲) ارائه‌شده است. نتایج حاصل، افزایش نرخ صحت در مدت‌زمان اجرای طولانی شبکه عصبی خود رمزگذار عمیق بر روی سناریوی استخراج خودکار ویژگی‌ها از مجموعه داده تولیدشده را نسبت به سناریوی استخراج دستی ویژگی‌ها نشان می‌دهد.



شکل (۱۱): نرخ صحت شبکه عصبی خود رمزگذار با دو سناریوی استخراج ویژگی دستی و خودکار پیشنهادی در محدوده زمانی ۱۰ ثانیه تا ۱۲۰ ثانیه



شکل (۱۲): نرخ صحت شبکه عصبی خود رمزگذار با دو سناریوی استخراج ویژگی دستی و خودکار پیشنهادی از دوره ۱۰ تا ۱۰۰ تکرار

۴-۳-۳- تحلیل نتایج

نتایج حاصل از پیاده‌سازی شبکه عصبی خود رمزگذار عمیق نشان از عملکرد مناسب این شبکه در دو سناریوی استخراج دستی و استخراج خودکار ویژگی‌ها را در سنجه‌های نرخ صحت، نرخ تشخیص و میانگین هارمونی و نرخ مثبت نادرست با کمترین زمان تشخیص دارد. بالاترین نرخ صحت، نرخ تشخیص، میانگین هارمونی و کمترین میزان مثبت نادرست با به‌کارگیری شبکه عصبی خود رمزگذار عمیق با سناریوی استخراج ویژگی خودکار به‌دست آمد. بالاترین میزان دقت نیز با استفاده از الگوریتم جنگل تصادفی و سریع‌ترین زمان تشخیص با مقدار ۲,۱۰۲ ثانیه توسط شبکه عصبی خود رمزگذار عمیق با سناریوی استخراج ویژگی دستی حاصل‌شده است. با بررسی کلی نتایج می‌توان توانایی شبکه عصبی خود رمزگذار عمیق با سناریوی استخراج ویژگی خودکار را در تشخیص دامنه‌های بدخواه و تولیدشده توسط الگوریتم‌های تولید دامنه مشاهده کرد. این شبکه عصبی، عملکرد مناسبی در تفکیک دامنه‌های سالم از دامنه‌های بدخواه و تولیدشده توسط الگوریتم‌های تولید دامنه را ارائه می‌دهد.

با توجه به جدول (۴)، معیار زمان تشخیص که در این مقاله به‌منظور نزدیک شدن به تشخیص برخط دارای اهمیت زیادی است، باوجوداینکه روش شبکه عصبی خود رمزگذار عمیق با سناریوی استخراج ویژگی دستی، دارای نتایج پایین‌تری نسبت به روش استفاده از شبکه عصبی خود رمزگذار عمیق با سناریوی استخراج ویژگی خودکار است ولی به‌دلیل اینکه در سناریوی استخراج ویژگی دستی، ویژگی‌های کمتری (۱۶ ویژگی) در مقابل سناریوی استخراج ویژگی خودکار (۷۳ ویژگی) به شبکه خود رمزگذار وارد می‌شود مدت‌زمان پردازش کمتر شده و مدت‌زمان تشخیص این روش بسیار سریع‌تر بوده و در مدت‌زمان ۲/۱۰۲ ثانیه توانایی تشخیص دامنه‌های بدخواه و ناسالم از دامنه‌های سالم را دارد. همچنین نرخ تشخیص قابل قبولی برابر با ۹۱/۲۹٪، نرخ دقتی برابر با ۹۲/۳۲٪ و نرخ مثبت نادرست پایینی برابر با ۰/۰۲۴ ارائه می‌دهد که می‌تواند برای استفاده در سامانه‌های تشخیص برخط شبکه‌های بات و تشخیص دامنه‌های تولیدشده توسط شبکه‌های بات امیدوارکننده باشد.

۵- نتیجه‌گیری و کارهای آینده

در سال‌های اخیر، شبکه‌های باتی که از الگوریتم‌های تولید دامنه برای ارتباط مدیر بات با سرویس‌دهنده‌های فرمان و کنترل خود استفاده می‌کنند افزایش چشمگیری داشته‌اند. کارهای بسیار زیادی در تشخیص دامنه‌های بدخواه تولیدشده توسط الگوریتم‌های تولید دامنه شبکه‌های بات، انجام‌یافته است، بااین‌حال، استخراج ویژگی از مجموعه داده‌ها در ارتباط با شناسایی دامنه‌های بدخواه به‌کار گرفته‌شده توسط شبکه‌های بات دارای اهمیت زیادی است. در این مقاله از دو سناریوی استخراج ویژگی دستی و خودکار بر روی مجموعه داده‌های متشکل از انواع دامنه‌های سالم و بدخواه استفاده شده است. همچنین شبکه عصبی خود رمزگذار عمیق برای تشخیص دامنه‌های بدخواه از دامنه‌های سالم به‌کار گرفته‌شده و میزان تشخیص دامنه‌های موجود بر اساس سنجه‌های مختلف ارزیابی شده است. نتایج آزمایش‌ها نشان داد که شبکه عصبی خود رمزگذار عمیق علی‌رغم دارا بودن نرخ دقت پایین‌تر نسبت به الگوریتم جنگل تصادفی، دارای نرخ صحت، نرخ تشخیص و میانگین هارمونی بالاتری است و همچنین نرخ مثبت‌کاذب و زمان تشخیص پایین‌تری دارد. با توجه به زمان تشخیص پایین‌تر و سرعت بیشتر شبکه عصبی خود رمزگذار عمیق نسبت به سایر الگوریتم‌ها می‌توان از این روش برای تشخیص برخط شبکه‌های بات استفاده‌کننده از دامنه‌های مخرب استفاده کرد.

در کارهای آینده می‌توان با استفاده از سایر روش‌های یادگیری عمیق و ترکیبی از روش‌های یادگیری و روش‌های انتخاب ویژگی‌های مناسب، دامنه‌های بدخواه و مشکوک تولیدشده توسط الگوریتم‌های تولید دامنه را مورد ارزیابی قرارداد.

۶- مراجع

[1] M. Asadi, S. Parsa, M. A. Jabraeil Jamali, and V. Majidnezhad, "P2P Botnet detection Using Deep Learning method," Journal of Electronical & Cyber Defence, vol. 8, no. 2, 2020. (In Persian)

[2] R. Jalaei and M. R. Hasani Ahangar, "An Analytical Survey on Botnet and Detection Methods," Journal of Electronical & Cyber Defence, vol. 4, no. 4, 2017. (In Persian)

[3] V. Mohammadi and A. Rezaee, "Botnets Detection by Analyzing Network Traffic Group Activities and Unsuccessful Responses," Passive Defense Quarterly, vol. 7, no.3, 2016. (In Persian)

[4] D. K. McGrath and M. Gupta, "Behind Phishing: An Examination of Phisher Modi Operandi," First USENIXWorkshop on Large-Scale Exploits and Emergent Threats, LEET '08, San Francisco, CA,USA, 2008.

[5] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Exposure: Finding Malicious Domains Using Passive DNS Analysis," Network and Distributed System Security Symposium, NDSS 2011, San Diego, CA, USA, 2011.

[6] J. Ma, L. K. Saul, S. Savage, and G.M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 2009.

[7] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware," 21st USENIX Security Symposium, Bellevue, WA, USA, 2012.

[8] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," 10th annual conference on Internet measurement MC '10, 2010.

[9] R. R. Curtin, A. B. Gardner, S. Grzonkowski, A. Kleymenov, and A. Mosquera, "Detecting DGA domains with recurrent neural networks and side information," 14th International Conference on Availability, Reliability and Security, 2019.

[10] N. Davuth and S-R. Kim, "Classification of malicious domain names using support vector machine and bi-gram method," J. Secur. Appl., vol. 7, pp. 51–58, 2013.

[11] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," arXiv Prepr, arXiv:1611.00791, 2016.

[12] K. Demertzis and L. Iliadis, "Evolving Smart URL Filter in a Zone-Based Policy Firewall for Detecting Algorithmically Generated Malicious Domains," Lecture Notes in Computer Science, pp. 223–233, 2015.

[13] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: DGA-Based Botnet Tracking and Intelligence," Lecture Notes in Computer Science, pp. 192–211, 2014.

[14] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting Algorithmically Generated Domain-Flux Attacks With DNS Traffic Analysis," IEEE/ACM Transactions on Networking, vol. 20, no. 5, pp. 1663–1677, 2012.

F. Ren, Z. Jiang, X. Wang, and J. Liu, "A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network," Cyber security, vol. 3, no.1, pp. 1-13, 2020.

[15] J. Hagen and S. Luo, "Why domain generating algorithms (dgas)," Trend Micro, Retrieved March, vol. 25, 2017.

[16] C. E. Shannon, "A Mathematical Theory of Communication," 2009.

[17] S. Douzi, M. Amar, and B. El Ouahidi, "Advanced Phishing Filter Using Autoencoder and Denoising Autoencoder," International Conference on Big Data and Internet of Thing – BDIOT'17, 2017.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv Prepr, arXiv:1412.6980, 2014.

[19] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.

[20] J. Wang, H. He, and D. V. Prokhorov, "A Folded Neural Network Autoencoder for Dimensionality Reduction," Procedia Computer Science, vol. 13, pp. 120–127, 2012.

[21] M. Abadi et al., "Tensorflow: a system for large-scale machine learning," OSDI, vol. 16, pp. 265–283, 2016.

[22] Chollet F., Keras, Accessed 2017-05-28. [Online]. Available: https://github.com/fchollet/keras

[23] Alexa Top 1 Million Sites: The Alexa Top Sites web service provides access to lists of websites ordered by Alexa Traffic Rank. (https://www.kaggle.com/ cheedcheed/top1m)

[24] Bambenek Consulting provided malicious algorithmically generated domains.(http://osint.bambenekconsulting.com /feeds/dga-feed.txt)

[25] 360 Lab DGA Domains: A collection of domains generated by DGA and it is maintained by 360-a Chinese security vendor. (https://data.netlab.360.com/feeds/dga/ dga.txt)

Journal of Electronical & Cyber Defence
Vol. 9, No. 1, 2021, Serial No. 33

Detecting of Botnets' Malicious Domains with Deep Autoencoder Neural Network

M. Asadi, S. Parsa*, V. Vosoghi

*Iran University of Science and Technology, Tehran, Iran

(Received: 02/04/2020, Accepted: 05/08/2020)

ABSTRACT

Botnet is a group of hosts infected with the same malicious code and managed by an attacker or Botmaster through one or more command and control (C&C) servers. The new generation of Botnets generates C&C domain name server's list dynamically. This dynamic list created by a domain generation algorithm helps an attacker to periodically change its C&C servers and prevent their addresses from being blacklisted. Each infected host generates a large number of domain names using a predefined algorithm and attempts to map them to their corresponding addresses by sending queries to the domain server. In this paper, the deep autoencoder neural network is used to identify domains without any knowledge of their generating algorithm, and the performance of the proposed method is compared with the performance of machine learning algorithms. Initially, a new dataset is created by combining a data set with normal domains and two datasets containing malicious and abnormal domains and both manual and automated methods are used to extract the features of the new dataset. Deep autoencoder neural network is applied to new and pre-processed datasets and the results are compared with machine learning algorithms. Based on the obtained results, it is possible to identify the malicious domains generated by domain generating algorithms using the deep autoencoder neural network with a higher speed and an accuracy rate larger than 98.61%.

Keywords: Botnet, Domain Generation Algorithms (DGAs), Feature Extraction, Deep Neural Network, Deep Autoencoder Neural Network

* Corresponding Author Email: Parsa@iust.ac.ir