Journal of Electronical & Cyber Defence
Vol. 9, No. 3, 2021, Serial No. 35

Presentation of a New Solution to Botnet Detection in a Markov Chain-Based Network

A. Ezzatneshan[1], S. R. Kamel Tabbakh Farizani[2*], M. Kheirabadi[3], R. Ghaemi[4]

* Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

(Received: 01/11/2020, Accepted: 25/02/2021)

ABSTRACT

Available botnets currently cover a wide range of Internet shipments. Use the net to access the network from infected computers connected to the Internet, remotely. Using research in this field is done based on the signatures with the result of the discovered results, anomalies, traffic behavior, and existing addresses. This method has not been able to detect a high rate at the moment, which is especially useful when it performs its main behavior, or these are methods that have already been forgotten due to need for memory. It is so great that it is practically impossible to do. The purpose of this study is to propose the construction to perform the identification operation, which is presented in this study with Markov chain and without the use of memory because Markov chain in this study does not require storage memory and does not exist based on behavioral analysis. The proposed method is able to perform useful behaviors using incorrect results of the operation better than the previous solutions, because if it examines the form you need, if such conditions do not exist, it will cause a computational overhead. In this research, various criteria such as medium circuit lines, accuracy and precision under consideration are captured, and in other of these proposed methods, as more possible than other existing methods, it is better if performed.

Keywords: Markov Chain, Botnet Discovery, Network Flow, Feature Extraction.

* Corresponding Author Email: mohammadengkhanjani@gmail.com

نشریه علمی "پدافند الکترونیکی و سایبری"
سال نهم، شماره ۳، پاییز ۱۴۰۰، ص ۷۱- ۵۹

علمی- پژوهشی

ارائه روشی نوین جهت شناسایی بات‌نت‌ها در شبکه مبتنی بر زنجیره مارکوف

عزیز عزّت نشان[۱]، سیّدرضا کامل طبّاخ فریضنی[۲*]، مریم خیرآبادی[۳]، رضا قائمی[٤]

۱-گروه مهندسی کامپیوتر، واحدنیشابور، دانشگاه آزاد اسلامی، نیشابور، ایران ۲- گروه مهندسی کامپیوتر، واحدمشهد، دانشگاه آزاد اسلامی، مشهد، ایران
۳- گروه مهندسی کامپیوتر، واحدنیشابور، دانشگاه آزاد اسلامی، نیشابور، ایران ٤- گروه مهندسی کامپیوتر، واحد قوچان، دانشگاه آزاد اسلامی، قوچان، ایران
(دریافت: ۱۳۹۹/۰۸/۱۱، پذیرش: ۱۳۹۹/۱۲/۰۷)

چکیده

بات‌نت‌ها در حال حاضر طیف وسیعی از حملات اینترنتی را تشکیل می‌دهند. بات‌نت‌ها، شبکه‌ای از کامپیوترهای آلوده متصل به اینترنت، با کنترل از راه دور می‌باشند. تاکنون تحقیقات زیادی در این زمینه انجام‌شده است که بر اساس امضاهای بات‌نت‌های کشف شده، ناهنجاری‌ها، رفتار ترافیکی، آدرس‌ها است. این روش‌ها تاکنون نتوانسته‌اند نرخ کشف بالایی را داشته باشند مخصوصاً برای بات‌نت‌هایی که در شرایط خاصی رفتار اصلی خود را بروز می‌دهند و یا این روش‌ها می‌بایست برای مقایسه گذشته بات را به‌طور کامل به خاطر بسپارند که این در مواردی نیازمند به حافظه بسیار بزرگی هست که در عمل غیرممکن می‌شود. هدف از این تحقیق پیشنهاد ساختاری برای انجام عملیات شناسایی است که این کار در این تحقیق مبتنی بر زنجیره مارکوف ارائه شده است و سعی بر عدم استفاده از حافظه است. زنجیره مارکوف ارائه شده در این تحقیق نیازمند به حافظه نگهداری نیست و بر اساس تحلیل رفتاری می باشد. روش پیشنهادی قادر است تا رفتارهای بات‌نت‌ها را با بررسی ناحیه رفتاری، بهتر از راهکارهای گذشته بررسی نماید که بدین شکل نیازمند به بررسی کل جریان نیست بلکه نقاط خاصی بررسی می‌شوند که این باعث کاهش سربار محاسباتی می‌شود. در این تحقیق معیارهای مختلفی همچون خطای میانگین مربعات، دقت و صحت مورد بررسی قرار گرفت و در تمامی این موارد روش پیشنهادی به‌صورت قابل ملاحظه‌ای بهتر از باقی روش‌های مورد مقایسه عمل نمود.

کلیدواژه‌ها: زنجیره مارکوف، کشف بات‌نت، جریان شبکه، استخراج ویژگی

۱- مقدمه

در شبکه، داشتن ساختاری مطمئن که جریانات مشکوک را شامل نشود به‌طورکلی احساس می‌شود زیرا امروزه بسیاری از حملاتی که صورت می‌گیرند به‌صورت جریاناتی وارد سیستم کاربر شده و صدماتی را به آن وارد می‌کنند. در واقع، داشتن یک زیرساخت امن نیاز اولیه برای حفاظت از هویت کاربران و اطلاعات است. تحت شعاع قرار دادن دسترس‌پذیری برخی از سرویس‌های اینترنتی به‌واسطه حمله‌هایی که مانع از وجود یک ارتباط پایا میان کاربران و این سرویس‌ها می‌گردد، می‌تواند تلفات اقتصادی هنگفتی را به بار آورده و همچنین تهدیدی برای امنیت ملی و سلامت جامعه است [۱ و ۲].

در شبکه، عامل ایجادکننده و تشخیص ترافیک ایجاد شده توسط نرم‌افزارهای کاربردی بسیار حائز اهمیت است. به‌طوری‌که می‌تواند اشراف ما را به رفتار بات‌ها به‌طور قابل‌ملاحظه‌ای افزایش دهد. در حال حاضر به‌طور گسترده از روش DPI[1] جهت کشف رفتار بات‌ها استفاده می‌شود. اما DPI دو اشکال عمده دارد که نمی‌تواند در تمامی موارد، انتظار کاربرانش را برآورده سازد. اولین مسئله پردازش بالا و هزینه آن است به‌طوری‌که با افزایش حجم ترافیک در شبکه عملاً نمی‌توان DPI را به شکل بلادرنگ استفاده نمود. از طرف دیگر روش‌های رمزنگاری محتوا، سبب می‌شود این روش به‌شدت تحت تأثیر قرارگرفته و کارایی لازم را نداشته باشد. از این‌رو محققان به دنبال روشی هستند که بتوانند ترافیک رمز شده در شبکه را به شکل بلادرنگ مورد تحلیل قرار دهند [۳]. اکثر روش‌های مبتنی بر تشخیص رفتار شبکه از دو نوع داده شامل ورودی بسته‌ها و جریان شبکه برای الگوریتم تشخیص استفاده می‌نمایند.

در برخی از کارهای گذشته، به شناسایی مبتنی بر میزبان اشاره شده است. این روش گرچه می‌تواند از انتشار بات‌ها و ارتباط آن‌ها جلوگیری کند اما به‌دلیل آنکه عموماً سازمان‌ها نمی‌توانند این سیاست امنیتی را در تمام سازمان خود اجرا نمایند، ناکارآمد عمل خواهد کرد. همچنین در برخی از منابع به شناسایی مراکز فرماندهی و کنترل بر اساس ترافیک DNS اشاره شده است. این روش گرچه در نسل‌های قبلی بات‌ها که مبتنی بر پروتکل IRC[2] بودند [۴] و الگوریتم تولید دامنه [۵] چندان قوی نداشتند، کارآمد عمل می‌کرد اما امروزه، با توجه به سازوکار رمزنگاری بات‌ها و الگوریتم‌های تولید دامنه بسیار قدرتمند،

*رایانامه نویسنده مسئول: mohammadengkhanjani@gmail.com

[1] Deep Packet Inspection

[2] Internet Relay Chat

ناشر: دانشگاه جامع امام حسین (ع)

تقریباً ناکارآمد خواهد بود. بنابراین جهت شناسایی بات‌ها نیازمند یک روش مستقل از پروتکل و رمزنگاری خواهیم بود که بر اساس همبستگی جریان شبکه اقدام به شناسایی کند.

بنا بر بررسی‌های انجام‌شده توسط شرکت امنیتی [۶] Marshal، هر سیستم آلوده بات‌نت، ۶۰۰۰۰۰ هرزنامه در روز ارسال می‌کند. بات‌نت‌های Xarvester و Rustock قوی‌ترین منتشرکنندگان هرزنامه در دنیا هستند و قادر به ارسال ۲۵۰۰۰ پیام در هر ساعت، ۶۰۰۰۰۰ پیام در هر روز و ۴/۲ میلیون پیام در هر هفته هستند، که تنها همین موضوع ضرورت بررسی انواع روش‌های کشف بات‌نت را ایجاد می‌کند. ازاین‌رو می‌بایست راهکاری را ارائه نمود که بتوان حملات بات‌نت را با دقت بالایی تشخیص داد. ازاین‌رو در این تحقیق راهکاری در این زمینه ارائه‌شده است که بتواند بات‌نت‌ها را با دقت بالایی کشف نماید و برای این کار از زنجیره مارکوف کمک گرفته‌شده است.

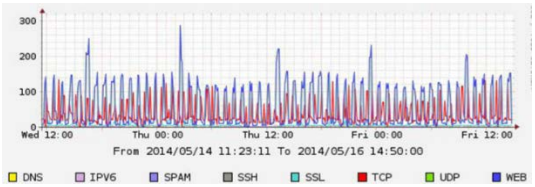
در ادامه ابتدا مفاهیم پایه بیان می‌شود و در ادامه آن نیز مروری بر کارهای گذشته صورت می‌گیرد. در قسمت بعد از آن روش پیشنهادی بیان می‌شود و در ادامه این روش مورد بررسی قرار می‌گیرد و در انتها جمع‌بندی کلی از کار بیان می‌شود و پیشنهادهایی برای کارهای آتی بیان می‌شود.

۲- مفاهیم اولیه و کارهای مرتبط

۲-۱- بات‌نت

هر بات‌نت شامل گروهی از ماشین‌های آلوده به کد مخرب یکسان است که توسط مهاجم و از طریق یک کانال فرمان و کنترل هدایت می‌شوند. بات بدافزاری است که رفتارهای بدخواهانه خود را بنا بر دریافت دستورهایی که توسط یک عنصر مرکزی فرستاده شده‌اند به انجام می‌رساند [۷]. بدافزار، نرم‌افزاری است که انگیزه تولید آن نفوذ، تغییر و یا تخریب نرم‌افزارهای دیگر بدون اطلاع و خواست کاربر آن‌ها است. به مجموعه‌ای از بات‌ها که با هم هماهنگ شده‌اند و توسط یک فرماندهی واحد (بات مستر) هدایت می‌شوند، شبکه بات می‌گویند. آنچه می‌تواند بات را به‌عنوان یک بدافزار از سایر گونه‌های بدافزارها متمایز کند هماهنگی بات‌ها و دستورپذیری آن‌ها از یک عنصر یکسان است. از دیگر تهدیداتی که می‌تواند توسط شبکه‌های بات ایجاد گردد سرقت اطلاعات، دست‌کاری نظرسنجی‌های اجتماعی، مصرف منابع کامپیوترهای آلوده به بات است. به‌طور مثال اخیراً استفاده از منابع بات جهت پردازش و استخراج بیت کوین، ارسال هرزنامه، حمله‌های منع سرویس توزیع شده، سرقت اطلاعات محرمانه و همچنین آلوده نمودن کامپیوترهای دیگر و گسترش در شبکه بوده است [۸]. گستردگی شبکه‌های بات و حضور تعداد زیادی از کامپیوترهای آلوده، علت اصلی خطرناک بودن شبکه‌ها است. در نتیجه حمله‌کننده می‌تواند از پهنای باند، قدرت ذخیره‌سازی و پردازش تعداد زیادی از کامپیوترها بهره بگیرد و به همان نسبت تهدید بزرگ‌تری ایجاد کند. با توجه به اهمیت موضوع بات‌نت، امروزه بیش‌از پیش ضرورت ایجاد سامانه‌ای برای کشف کامپیوترهای آلوده به بات مورد توجه قرارگرفته است. در یک شبکه بات سه وظیفه وجود دارد [۹]: زامبی، Relay و C&C Server.

بات‌ها دارای رفتارهایی هستند که خود را از دید سامانه‌های تشخیص، پنهان نمایند. شکل (۱) ترافیک بات سلیتی[1] را نشان می‌دهد که جهت ارتباط با مرکز فرماندهی و کنترل خود، ترافیکش را در پروتکل‌های مختلف شبکه تونل کرده است.



شکل (۱): ارتباط تونل بات سلیتی با پروتکل‌های ارتباطی مختلف [۹].

همان‌طور که در این شکل مشاهده می‌شود سلیتی از پروتکل‌های مختلف شبکه جهت ارتباط با مرکز فرماندهی و کنترل خود آن‌هم به‌صورت رمز شده استفاده می‌کند.

استخراج ویژگی فنی است جهت حذف ویژگی‌های نامرتبط و زائد و انتخاب بهترین زیرمجموعه از ویژگی‌ها که خصوصیات بهتری از الگوهای متعلق به کلاس‌های متفاوت را تولید می‌کند. به بیان ساده‌تر روشی که در آن ویژگی‌هایی از جریان شبکه را استخراج می‌کنیم تا بتوانیم تولیدکننده جریان‌ها را از هم تمیز دهیم. به‌طور کلی استخراج ویژگی باعث می‌شود که حجم محاسبات کاهش یابد و همچنین در مواردی باعث کاهش نویز نیز می‌شود که خود درصد خطا را کاهش می‌دهد و باعث افزایش در دقت پیش‌بینی‌ها می‌شود.

روش‌های تشخیص بات‌نت می‌توانند دو سطح مختلف از تحلیل همبستگی شامل سطح انفرادی و سطح گروهی را به‌کار گیرند [۱۰].

در تحلیل سطح انفرادی، تمرکز روش‌های تشخیص بات‌نت بر روی شناسایی هر میزبان آلوده به بات به‌صورت انفرادی در شبکه هست، بدون این‌که به رفتار میزبان‌های آلوده به بات دیگر توجه نماید. این روش‌ها دارای این مزیت هستند که قادرند حتی یک میزبان آلوده به بات را در شبکه تحت نظارت تشخیص دهند [۱۱]. تحلیل سطح انفرادی معمولاً از طریق انطباق فعالیت‌های مشاهده شده با الگوهای شناخته شده موجود در پایگاه داده انجام

[1] sality

می‌شود. بنابراین نیاز به دانش قبلی از بات‌نت‌ها دارد. در مقابل روش‌های سطح گروهی، یک بات به‌تنهایی عمل نمی‌کند بلکه عملکرد اصلی ناشی از ترکیب چندین بات هست که بدین صورت تشخیص آن‌ها را کمی دشوار می‌کند ولی از طرفی برای خود بات‌ها نیز این عمل کار دشواری است زیرا اگر در میان راه یکی از این بات‌ها شناسایی شوند کل عملکرد از کار می‌افتد زیرا عملکرد اصلی یا حمله بات ناشی از ترکیب عملکرد تمامی این بات‌ها هست. برای تشخیص حتی در سطح گروهی نیز می‌توان به همان شکل انفرادی عمل نمود با این تفاوت که به دنبال یک حمله کلی نبود و با مشاهده کوچک‌ترین خطرپذیری حمله، آن جریان به‌عنوان بات شناخته شود که بدین شکل می‌توان از حمله گروهی جلوگیری نمود.

۲-۲- ماتریس تصادفی

ماتریس‌های تصادفی در علومی همچون کامپیوتر، شیمی، اقتصاد و غیره مورد استفاده قرار می‌گیرد. در حقیقت ماتریس تصادفی می‌تواند جهت پیش‌بینی مسائل متفاوت در حوزه‌های علمی مختلف استفاده شود. ماتریس تصادفی یا ماتریس مارکوف ماتریسی است که جهت تشریح انتقال حالات در زنجیره مارکوف مورد استفاده قرار می‌گیرد. ماتریس تصادفی را با نام‌های ماتریس گذار و ماتریس احتمال نیز می‌شناسیم.

۲-۳- زنجیره مارکوف

زنجیره مارکوف مدلی تصادفی برای توصیف یک توالی از رویدادهای احتمالی است که در آن احتمال هر رویداد از رویداد دیگر کاملاً مستقل هست. زنجیره مارکوف یک فرایند تصادفی بدون حافظه است بدین معنی که توزیع احتمال شرطی حالت بعد تنها به حالت فعلی بستگی دارد و مستقل از گذشته آن است [۱۲].

تغییرات حالات سیستم انتقال نام دارند و احتمال‌هایی که به این تغییر حالت‌ها نسبت داده می‌شوند احتمال انتقال نام دارند. یک فرایند مارکوف با یک فضای حالت شمارا، یک ماتریس گذار برای توصیف احتمال‌های هر انتقال و یک حالت اولیه (یا توزیع اولیه) در فضای حالت مشخص می‌شود [۱۳].

۲-۴- کارهای مرتبط

Zeng و همکارانش روشی ترکیبی بر مبنای مشاهدات در سطح شبکه و میزبان ارائه دادند، چارچوب کاری آن‌ها بدین شکل است که ابتدا جریان موجود در شبکه را بررسی و تحلیل‌و بررسی[1] می‌کند و آن‌هایی که ترافیک مشابهی دارند را کشف می‌کنند [۱۴].

Strayer و همکارانش استفاده از فن یادگیری ماشین[2] را برای کشف ترافیک IRC آلوده پیشنهاد کردند. کاری که آن‌ها انجام دادند در دو لایه خلاصه می‌شود، در لایه اول ترافیک IRC و غیر IRC از هم جدا می‌شوند و سپس بین ترافیک بات‌نت و ترافیک قانونی IRC تفکیک انجام می‌شود [۱۵].

T.Ha و همکارانش روی kademila (نوعی بات‌نت P2P) تمرکز کردند. آن‌ها دیدگاه تازه‌ای را در رابطه با کشف بات‌نت‌ها با استفاده از نقاط راهبردی بیان کرد. آن‌ها عامل‌هایی مثل درجه مرکزیت، درجه تراکم و مرکزیت بردار Eigen و مرکزیت بر پایه مسیریابی[3] معرفی کردند. به‌عنوان نمونه مرکزیت بردار Eigen می‌گوید تمامی اتصالات به یک میزبان خاص به‌اندازه هم مهم نیستند، بلکه آن‌هایی مهم‌تر هستند که به گره‌های مهم‌تر در شبکه وصل شده‌اند. با مشخص شدن این عامل‌ها در شبکه می‌توان انتخاب‌های بهتری برای نقاط نظارتی در شبکه انجام داد، چون همان‌طور که قبلاً نیز گفته شد امکان نظارت بر کل شبکه وجود ندارد. همچنین می‌توان نقاط گلوگاه را که در حیات شبکه بات‌ها تأثیر به‌سزایی دارد را تعیین کرد [۱۶].

در [۱۷]، بر اساس روش مبتنی بر میزبان عمل می‌کند. این روش به بررسی مولفه‌ها و آرگومان‌های توابع فراخوانی شده سیستمی که حتی می‌توانند رمز شده باشند، اقدام می‌کند. این روش الهام بخش روش [۱۸ و ۱۹] شده است اما با این تفاوت که برخلاف الگو استینسون، بر روی آرگومان‌ها تمرکز ندارد بلکه بر روی یک سری از توابع فراخوانی شده سیستمی به‌خصوص که برای عملکردهای مغرضانه به‌کار می‌روند، تأکید دارد. در این زمینه روش‌هایی از تجمیع گزارش بهره می‌برند. مانند فن‌های به‌کار گرفته‌شده توسط [۲۰ و ۲۱] که روی پروتکل‌های مبتنی بر گفتگوی اینترنتی کار می‌کنند. همچنین فن‌هایی که توسط [۲۳- ۲۱] ارائه شده است و بر روی ساختارهای متمرکز کارکرده است. این روش مبتنی بر میزبان، برای فرایند تحلیل از یک محافظ مجازی استفاده می‌کند تا به تجمیع گزارش‌های میزبان برای کشف بات بپردازد.

در این روش با کمک طراحی یک دنبال‌کننده واسط برنامه‌نویسی که با توجه به [۲۴] طراحی‌شده است، به بررسی و کشف فراخوانی توابعی که توسط بات‌ها برای اعمال مخرب و مغرضانه خود، بارها استفاده می‌شود، می‌پردازد. در روشی دیگر، [۲۱] با به‌کارگیری روش مبتنی بر میزبان، سعی دارد از روش تجمیع گزارش‌ها به تشخیص یک یا مجموعه‌ای از بات‌ها بپردازد. ایده تجمیع گزارش‌ها در [۲۵] مطرح‌شده است که ادعا کرده است که با تجمیع گزارش‌ها از منابع مختلف، با دقت بهتری

[1] Analyze

[2] Machine Learning
[3] Routing

می‌توان به تشخیص بات‌ها پرداخت. روش بات سوات [۱۷] بر روی رفتار بات بر اساس نظارت بر برنامه‌های اجرا شده در لیست کتابخانه Win32 موجود در میزبان کار می‌کند. بخش کلیدی این روش به‌کارگیری Detours است که به کمک آن فرایند نظارت بر توابع فراخوانی واسط برنامه‌نویسی را انجام می‌دهد. این روش‌ها همگی مبتنی بر میزبان هستند و قابلیت به‌کارگیری در کلیه سیستم‌های عامل و معماری‌های شبکه را ندارند.

در [۲۶]، تجمیع گزارش‌های به‌دست‌آمده از پویش ترافیک ورودی و خروجی، انجام می‌شود. این روش فقط در مقیاس شبکه به تشخیص می‌پردازد و هیچگونه حساسیتی برای کشف ثبت‌کننده‌های کلید و یا فرایندهای پاک کردن فایل‌ها در مقیاس میزبان ندارد.

در [۹]، Gu و همکاران یک روش مبتنی بر خوشه‌بندی برای تشخیص بات‌نت‌ها در مرحله حمله ارائه کرده‌اند. در این روش، ابتدا ترافیک ارتباطی مشابه و ترافیک بدخواهانه مشابه خوشه‌بندی شده و سپس یک همبستگی بین خوشه‌ای انجام می‌شود تا میزبان‌های دارای هر دو الگوی فعالیت بدخواهانه مشابه شناسایی شوند. روش فوق به‌صورت غیر بر خط عمل می‌کند که در سیستم‌های تشخیص بات‌نت یک ضعف عمده به‌شمار می‌آید. همچنین درصورتی‌که بات‌های عضو یک بات‌نت در مرحله حمله فعالیت بدخواهانه جدیدی را انجام دهند، این روش قادر به تشخیص آن بات‌نت نخواهد بود.

در [۲۷]، Wang و همکاران یک سیستم مبتنی بر رفتار برای تشخیص بات‌نت‌ها ارائه کرده‌اند که بر اساس فن‌های بازشناسی الگوی فازی است و از تحلیل سطح انفرادی استفاده می‌کند. ایده اصلی روش آن‌ها بر مبنای شناسایی نام‌های دامنه و آدرس‌های IP بدخواه مورد استفاده توسط بات‌نت هست که از طریق بازرسی جریان‌های شبکه حاصل می‌شوند. این روش از سه مرحله کاهش ترافیک، استخراج ویژگی و بازشناسی الگوی فازی تشکیل شده است. در ابتدا ترافیک شبکه وارد مرحله کاهش ترافیک شده و پس از پالایش به مرحله استخراج ویژگی تحویل داده می‌شوند. در نهایت، مرحله بازشناسی الگوی فازی، فعالیت‌های مرتبط با بات‌نت را بر اساس تعلق بردارهای ویژگی استخراج شده به چندین تابع عضویت تشخیص می‌دهند. این توابع عضویت شامل:

۱.تولید درخواست‌های DNS ناموفق

۲.مشابهت در فاصله‌های ارسال درخواست‌های DNS

۳.تولید اتصالات ناموفق شبکه‌ای

۴.داشتن اندازه بدنه یکسان در اتصالات شبکه‌ای آن‌ها می‌باشند.

در [۳۰] روش مبتنی بر شبکه است و از ساختار و پروتکل‌های سرورهای فرمان و کنترل مستقل هست. این روش، بات‌ها را بر اساس شباهت ترافیکی آن‌ها خوشه‌بندی می‌کند. پس از اتمام فرایند خوشه‌بندی، نحوه ارتباط خوشه‌ها با یکدیگر نیز مورد بررسی قرار می‌گیرد تا مشخص شود آیا ارتباطی بین فعالیت‌های مغرضانه خوشه‌ای با خوشه دیگر وجود دارد یا خیر. درصورتی‌که حملات از نوع ترکیبی و متنوع باشد، به‌دلیل استفاده از روش‌های مبتنی بر الگو، پس از خوشه‌بندی، احتمال عدم کشف بات‌ها بالا می‌رود.

۳- روش پیشنهادی

در این قسمت راهکاری برای شناسایی حملات بات‌نت‌ها معرفی می‌شود که قادر است تا بدون به خاطر سپردن گذشته و با کارایی بالا این حملات را شناسایی نماید.

روش پیشنهادی که در این قسمت برای شناسایی حملات بات‌نت‌ها بیان می‌شود شامل بخش‌های مختلفی هست که در ادامه هر یک از این بخش‌ها با جزئیات توضیح داده می‌شوند:

- استخراج ویژگی جریان‌ها: استخراج اطلاعات جهت تحلیل بیشتر
- انتخاب ویژگی‌های مؤثر: انتخاب ویژگی‌های با بیشترین تأثیرگذاری در جهت شناخت بات‌نت و غیر بات‌نت
- تجمیع جریان شبکه: در این مرحله دسته‌بندی درستی صورت می‌گیرد تا اینکه بتوان اطلاعات جریان‌های را بهتر بررسی نمود.
- استخراج ویژگی‌های خارج جریانی: به تحلیل رفتار ترافیک یک بات‌نت پرداخته می‌شود.
- حذف داده‌های نویزی: حذف داده‌های بی‌تأثیر که باعث ایجاد نویز و کاهش دقت می‌شوند.
- تولید الگو: برای ذخیره‌سازی و یا مقایسه
- مشخص ساختن مقدار آستانه: در جهت استفاده از آن برای شناسایی
- الگو کردن و استخراج الگوها: استخراج الگوهای برای شناسایی بات‌نت‌ها
- آموزش سیستم: آموزش سیستم در جهت شناسایی بات‌نت‌ها در ادامه
- بررسی جریانات جدید: شناسایی بات‌نت‌ها از جریان‌های سالم

۳-۱- استخراج ویژگی جریان‌ها

در این بخش از روش پیشنهادی که به‌طور مستقیم با کارت شبکه در ارتباط است ترافیک شبکه به‌صورت بلادرنگ دریافت می‌شود و مبتنی بر استاندارد جریان شبکه، مجموعه‌ای از بسته‌ها را به‌عنوان یک جریان در نظر می‌گیرد. به‌طور کلی ما در این تحقیق دو نوع اطلاعات ورودی داریم اطلاعاتی که با استفاده از ویژگی‌های مختلف کلاً درون جریان می‌توان استخراج کرد مانند آدرس مبدأ، پورت مبدأ، آدرس مقصد، پورت مقصد و پروتکل و دسته دیگری که از خارج جریان قابل استخراج می‌باشند مانند اندازه جریان، مدت‌زمان شروع تا پایان یک جریان و تناوبی بودن یک جریان که برای هر یک می‌بایست آماده‌سازی صورت گیرد در این مرحله ویژگی‌های دسته اول یا اطلاعات درون جریانی استخراج می‌شوند. این ویژگی‌ها شامل موارد مختلفی هست که در ادامه این روند تشخیص ویژگی‌های مؤثرتر معرفی می‌شوند و توضیح داده می‌شود که چطور ویژگی‌های با تأثیرگذار بالاتر شناسایی می‌شوند.

۳-۲- انتخاب ویژگی‌های مؤثر

برای استخراج ویژگی‌ها در این قسمت از الگوریتم شبکه عصبی که با استفاده از الگوریتم ژنتیک بهبود داده‌شده است استفاده می‌شود. الگوریتم ژنتیک، با قابلیت قابل‌توجه در استنتاج معانی از داده‌های پیچیده، می‌تواند برای استخراج الگوها و شناسایی روش‌هایی که آگاهی از آن‌ها برای انسان و دیگر فن‌های کامپیوتری بسیار پیچیده و دشوار است به‌کار گرفته شود. الگوریتم ژنتیک به‌عنوان یکی از ابزارهای داده‌کاوی می‌تواند برای طبقه‌بندی مورد استفاده قرار گیرد. در اینجا هدف استفاده از الگوریتم شبکه‌های عصبی پرسپترون که به‌وسیله الگوریتم ژنتیک بهینه‌شده، در جهت شناسایی ویژگی‌های مؤثرتر هست تا در نتیجه ویژگی‌های مؤثر در جهت شناسایی حملات را شناسایی نمود. در واقع در این قسمت از راهکار بیان شده در مقاله [۳۱] که توسط لدسما و همکارانش در سال ۲۰۰۸ معرفی شد.

شبکه‌های عصبی مصنوعی از رهیافت‌های الگوسازی با الهام از ویژگی‌های سیستم عصبی انسان هست که در حوزه‌های مختلف علم و مهندسی برای پیش‌بینی پدیده‌های پیچیده و غیرخطی مورد استفاده قرار می‌گیرند. ساختار شبکه‌های عصبی حداقل با داشتن سه لایه شامل لایه ورودی، حداقل یک لایه مخفی و یک لایه خروجی قابل تعریف هست. هر ورودی به شبکه عصبی به یک وزن سیناپسی ضرب شده و با یک بایاس جمع می‌شود. از الگوریتم‌های مختلف شبکه برای آموزش الگوی موجود بین داده‌ها استفاده می‌شود و از سه قسمت آموزش شبکه، اعتبارسنجی و آزمون شبکه چگونگی الگو دریافت شده قابل ارزیابی هست. الگوریتم پس انتشار به‌عنوان کارآمدترین الگوریتم برای شبکه‌های چند لایه پرسپترون پیش خوران[1] شناخته می‌شود و برای بهینه‌سازی وزن‌ها و بایاس شبکه استفاده می‌شود که باعث بهبود عملکرد شبکه بعد از هر دوره می‌شود. اکثر روش‌های مورد استفاده برای کاهش خطای شبکه عصبی از توابع گرادیان برحسب الگوریتم پس انتشار تبعیت می‌کنند. الگوریتم‌های مختلف بهینه‌سازی بر مبنای جمعیت ذرات نیز می‌توانند با شبکه عصبی ادغام‌شده و عملکرد شبکه را بهبود ببخشند.

تعیین ساختار شبکه از قدم‌های تأثیرگذار بر روی نحوه آموزش شبکه هست البته تعداد نورون بالا در شبکه پیچیدگی آن را افزایش می‌دهد و ممکن است شبکه دچار بیش برازش شود که قابلیت پیش‌بینی شبکه را تحت تأثیر قرار خواهد داد. به این جهت برای هر یک از الگوریتم‌های آموزشی تعداد نورون‌ها از یک تا ۱۰ افزایش یافتند. داده‌ها در محدوده ۱- تا ۱ عادی‌سازی[2] شدند و از تابع انتقالی تانژانتی- سیگموییدی[3] استفاده شد. از دو شاخص ارزیابی عملکرد شبکه خطای میانگین مربعات (MSE) و ضریب تبیین (R2) بهره گرفته شد. از رایج‌ترین الگوریتم‌های بهینه‌سازی که بر اساس دگرگونی بیولوژیکی به وجود آمده است، الگوریتم ژنتیک هست. جمعیت در الگوریتم ژنتیک شامل پاسخ‌های ممکن در فرم آرایه‌ای از کروموزوم‌ها باشد. وزن‌های شبکه درصورتی‌که توسط الگوریتم ژنتیک بهینه‌سازی می‌شوند بنابراین هر یک از جمعیت‌ها به‌صورت تصادفی به‌عنوان وزن شبکه معرفی شدند. تابع MSE به‌عنوان تابع هزینه معرفی شد و کروموزوم‌های جمعیت سپس برای رسیدن به کمترین تابع هزینه مرتب می‌شوند. تعداد مشخصی از اعضای بهتر بر اساس کمترین هزینه به نسل بعدی منتقل می‌شوند. در این مرحله سه اپراتور الگوریتم ژنتیک (انتخاب، تقاطع و دگرگونی) برای تولید جمعیت نسل بعدی فعال می‌شوند. این سیکل تا رسیدن به جواب مطلوب ادامه یافته تا وزن‌های مطلوب شبکه حاصل شوند.

۳-۳- تجمیع جریان شبکه

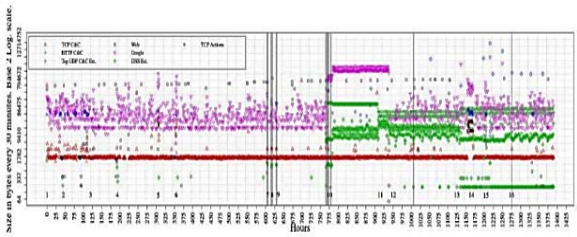
در این مرحله جریان خروجی از مرحله قبل را بهبود می‌دهد. در این مرحله قطعاً پورت مبدأ جز ویژگی‌های تأثیرگذار هست و ازاین‌رو سعی می‌شود تا این به‌طور قطع انتخاب شود و این قضیه با استفاده از پیشینه تجربی به‌دست آمده است. از میان جریان ورودی مورد بررسی با توجه به ویژگی‌های استخراج شده مرحله قبل و پورت مبدأ که قطعاً جز ویژگی‌های مؤثر خواهد بود، پورت مبدأ مشترک در این مرحله حذف می‌شوند که بدین شکل بتوان

[1] Feedforward Neural Network
[2] Normalize
[3] tansig

به ورودی‌های مناسب‌تری دست یافت. هدف از حذف پورت مبدأ این است که یک بات عمدتاً از پورت‌های متفاوت جهت ارتباط با مرکز فرماندهی و کنترل خود استفاده می‌کند. بنابراین خروجی این بخش مجموعه‌ای از جریان‌ها با ویژگی‌های مؤثر خواهد بود. در نتیجه به هر یک از این رکوردهای استخراج‌شده از جریان با ویژگی‌های مؤثر یک اتصال[1] می‌نامیم. این اتصال‌ها در مراحل بعدی کار مورد استفاده قرار می‌گیرند.

۳-۴- استخراج ویژگی‌های خارج جریانی

تحلیل رفتار ترافیک یک باتنت دارای سه قسمت فعالیت‌ها، اتصالات و الگوها هست. اکثر بات‌ها برای فعالیت‌های خود از چندین ماژول متفاوت استفاده می‌کنند که هریک می‌توانند الگوهای رفتاری مختلفی تولید کنند. برای تحلیل تمامی این رفتارها نیاز است زمان زیادی باتنت و ترافیک تولیدی آن را مورد بررسی قرار داد تا در این مدت باتنت تمامی الگوهای رفتاری خود را بروز دهد. که برای این کار رفتار بات‌ها برای ۵۷ روز مورد بررسی قرار گرفت که می‌توان نتیجه این بررسی را در شکل (۲) مشاهده نمود.



شکل (۲): رفتار استخراج‌شده از باتنت‌ها در طول ۵۷ روز.

در این شکل ستون عمودی اندازه ترافیک تولیدشده را نشان می‌دهد و همچنین ستون افقی نشان‌دهنده زمان است. خط قرمزرنگ نشان می‌دهد باتنت به‌طور پیوسته با یک حجم مشخص اقدام به تولید ترافیک TCP می‌کند. این وضعیت در سایر پروتکل‌ها نیز با کمی انحراف نشان داده‌شده است. بنابراین چارچوب پیشنهادی از سه ویژگی اندازه جریان، مدت‌زمان شروع تا پایان یک جریان و تناوبی بودن یک جریان جهت الگو کردن رفتار باتنت‌ها استفاده می‌نماید. این سه ویژگی در نتیجه بررسی‌های صورت گرفته و کمک گرفتن از مقاله [۳۲] به‌دست‌آمده است.

ازآنجایی‌که ویژگی تناوبی بودن راهکار مشخصی برای محاسبه ندارد. برای محاسبه این ویژگی راهکاری را بیان نمودیم که با استفاده از آن بتوان ترافیک بات‌ها را مورد تحلیل و بررسی قرار داد و حملات را شناسایی نمود.

برای بررسی تناوبی بودن رفتار بات‌ها بر اساس ویژگی مدت‌زمان جریان و حجم داده‌های انتقالی از رابطه (۱) استفاده می‌کنیم.

$$Correlation(flow) = \frac{(T_{flow} - \bar{T})(S_{flow} - \bar{S})}{\sqrt{(T_{flow} - \bar{T})^2} + \sqrt{(S_{flow} - \bar{S})^2}} \qquad (۱)$$

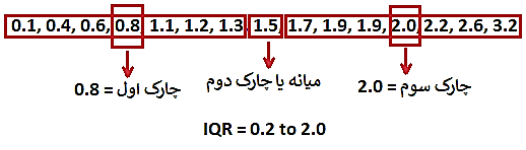
که در این رابطه T_{flow} زمان جریان، S_{flow} حجم جریان، $\bar{T}$ میانگین زمان ورودی جریان‌های قبلی و $\bar{S}$ میانگین حجم جریان‌های واردشده قبلی هست. با استفاده از این رابطه می‌توان تناوبی بودن رفتار بات‌ها را موردبررسی قرار داد. که با توجه به اینکه از قبل بررسی روی این قضیه صورت گرفته است جدولی استخراج‌شده است که با استفاده از آن بتوان نتایج را واضح‌تر موردبررسی قرار داد که در جدول (۱) قابل‌مشاهده هست. با توجه به اینکه معیار مشخص‌شده رنجی مابین ۰ تا ۱ دارد و اینکه هر چه به مقدار ۱ نزدیک‌تر باشد با توجه به رابطه (۱)، میزان همبستگی بیشتر می‌شود و یا تناوبی خیلی قوی می‌شود، مقدار این معیار را در بازه‌های مختلف مابین ۰ تا ۱ تقسیم‌بندی نمودیم. هر چه این معیار به صفر نزدیک باشد نشان‌دهنده این است که همبستگی ضعیف‌تری دارد و یا دارای تناوبی ضعیفی هست.

جدول (۱): دسته‌بندی جریان‌ها بر اساس رابطه همبستگی.

Correlation Result	برچسب اختصاص داده شده
صفر الی ۰/۱۹	تناوبی خیلی ضعیف
۰/۲ الی ۰/۳۹	تناوبی ضعیف
۰/۴ الی ۰/۵۹	تناوبی میانی
۰/۶ الی ۰/۷۹	تناوبی قوی
۰/۸ الی ۱	تناوبی خیلی قوی

۳-۵- حذف داده‌های نویزی

در روش پیشنهادی داده‌های نویزی باید حذف شوند که در این حالت باعث افزایش دقت می‌شود منظور از داده‌های نویزی، داده‌ای با رکوردهای نویزی هست که این رکوردها می‌بایست از جریان ورودی حذف شوند تا اینکه حجم محاسبات کاهش و دقت افزایش یابد. برای اینکه بتوان داده‌های نویزی به‌وسیله روش پیشنهادی کشف نمود جریان ورودی به دو قسمت در ابتدای کار تقسیم می‌شود و میانه آن‌ها مشخص می‌شود. بعد از تقسیم داده‌ها به دو قسمت مساوی، هر یک از این دو قسمت نیز میانه آن‌ها یافت می‌شود که در این حالت جریان ورودی به چهار قسمت به‌وسیله سه چارک تقسیم می‌شوند.



شکل (۳): مثالی از تقسیم داده‌ها در روش پیشنهادی

[1] Connection۱۱

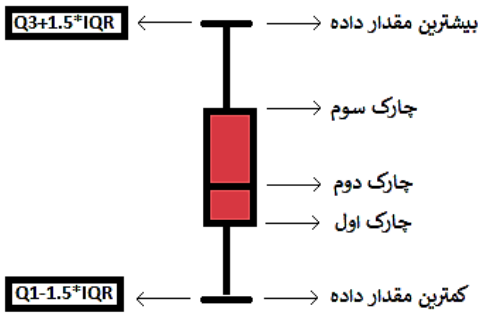
می‌توان در شکل (۳) مشاهده نمود که داده‌ها به قسمت‌های مساوی تقسیم‌شده‌اند و میانه آن ۱/۵ هست و در ادامه چارک‌های هر دو سمت این میانه نیز محاسبه‌شده است که ۲ و ۰/۸ می‌باشند.

برای اینکه بتوان داده‌های بهتری را از میان کل جریان‌ها استخراج نمود باید محدوده مؤثری را انتخاب کرد و اصطلاحاً مقدار بین چارکی انتخاب نمود. البته باید توجه کرد که در اینجا منظور این نیست که تمامی مقادیر که در این محدوده‌ها نیستند به‌عنوان داده‌های نویزی هستند بلکه منظور این است که محدوده‌های مؤثرتر شناسایی می‌شوند که با احتمال خیلی بالاتری بات رفتار اصلی خود را در آن‌ها نشان می‌دهد و در خارج از این محدوده‌ها احتمال بروز رفتار بات بسیار پایین‌تر است و این بدین معنی نیست که غیرممکن باشد بلکه تنها می‌توان گفت احتمال آن بسیار پایین است و در محدوده‌ای که ما انتخاب می‌کنیم اگر جریانی مخرب باشد با احتمال بسیار زیاد رفتار اصلی آن نشان داده می‌شود.

برای اینکه بتوان محدوده‌های انتخابی بین چارکی را محاسبه نمود از فرمول (۲) استفاده شده است.

$$IQR = Q_3 - Q_1 \qquad (۲)$$

برای درک بهتر این فرمول می‌توان شکل (۴) را مشاهده نمود. همان‌طور که مشهود است محدوده قرمزرنگ محدوده رفتار اصلی بات هست که در محدوده مؤثر شناسایی‌شده در روش پیشنهادی قرارگرفته است و محدوده خارج از این محدوده در نظر گرفته نمی‌شود



شکل (۴): نحوه محاسبه محدوده مؤثر بین چارکی

می‌توان مشاهده نمود که در اینجا با استفاده از فرمول (۲)، IQR محاسبه می‌شود که در این مثال این مقدار برابر است با ۲ - ۰/۸ = ۱/۲ و از طرفی محدوده برابر است با چارک اول منهای میانه ضربدر IQR و چارک سوم بعلاوه میانه ضربدر IQR که از این میان محدوده مشخص می‌شود (فرمول ۳) و به‌عنوان داده‌های موردنیاز این جریان در نظر گرفته می‌شود و بدین‌صورت می‌توان رفتار اصلی را شناسایی نمود زیرا بات‌ها همگی در ابتدا کار رفتار اصلی خود را نشان نمی‌دهند بلکه در اواسط جریان به رفتار اصلی خود می‌پردازند تا بدین شکل شناسایی را دشوار کنند. در شکل (۴) محدوده قرمز رنگ رفتار اصلی است که در دامنه مورد نظر قرارگرفته است و در نتیجه قطعاً در محدوده موردبررسی قرار می‌گیرد.

همان‌طور که در شکل (۴) مشاهده می‌شود داده‌هایی که خارج از محدوده فرمول (۳) هستند از مجموعه جریان‌های آموزشی حذف خواهند شد تا بتوانیم الگویی دقیق از رفتار بات بسازیم.

$$(utliers > Q_3 + M * IQR) or (outliers < Q_1 - M * IQR) \qquad (۳)$$

که در این فرمول M همان میانه هست و outliers همان محدوده خارج از محدوده بررسی هست که می‌بایست از جریان ورودی حذف شوند.

۳-۶- استخراج الگوهای رفتاری

در این مرحله الگوهای رفتاری برای تحلیل ترافیک شبکه به دست می‌آید. در این مرحله هدف استخراج اطلاعات برای تحلیلی بهتر است که از اطلاعات خارج جریانی برای این کار استفاده می‌شود. بنابراین با استفاده از ۳ ویژگی حجم جریان، زمان جریان و متناوب بودن، ترافیک ورودی از شبکه الگو می‌شود. ویژگی‌های اندازه برحسب بایت و بازه زمانی برحسب ثانیه است و می‌توان از هر جریان این ویژگی‌ها را استخراج کرد، ازاین‌رو این ویژگی‌ها را خارج جریانی نامیدیم. ویژگی تناوبی بودن به‌دلیل خاصیت خودکار بودن در اکثر بات‌نت‌ها وجود دارد و در روش‌های [۲، ۳ و ۴] از همین ویژگی جهت تشخیص بات‌نت استفاده‌شده است. اما لازم است در اینجا در مورد ویژگی سوم که تناوبی بودن یک جریان را نشان می‌دهد کمی بیشتر توضیح دهیم. تناوبی بودن هر جریان با توجه به جریان‌ها قبلی قابل‌محاسبه است. حداقل سه جریان در شبکه باید وجود داشته باشد تا بتوان در مورد تناوبی بودن جریان سوم نظر داد. به‌بیان‌دیگر برای محاسبه این ویژگی اختلاف‌زمان اتمام جریان اول تا شروع جریان دوم را محاسبه کرده و T1 می‌نامیم به همین ترتیب اختلاف‌زمان جریان دوم و سوم را محاسبه کرده و T2 می‌نامیم. برای بررسی تناوبی بودن از فرمول (۳) استفاده می‌شود.

این کار تا جایی که تمامی جریان‌ها موردبررسی قرار بگیرند ادامه می‌یابد و تمامی تناوب‌ها استخراج می‌شوند این بخش خود نیز در یک قالب برای بررسی ذخیره می‌شوند. همان‌طور که پیش‌تر نیز توضیح داده شد روش پیشنهادی با استفاده از دو دسته اطلاعات به شناسایی می‌پردازد که این بخش در واقع اطلاعات دسته دوم یا همان اطلاعات خارج جریانی هست.

۳-۷- مشخص ساختن مقدار آستانه

در این مرحله مقدار آستانه‌ای برای بررسی ویژگی‌های خارج جریانی مشخص می‌شود که همان مقادیر چارک‌ها هست که از کجای مقادیر به‌عنوان میانه یا چارک انتخاب شود. به‌طورکلی دو حالت برای انتخاب چارک‌ها وجود دارد که در این تحقیق استفاده شده است: ایستا و پویا.

در حالت آستانه ایستا، مقدار چارک‌ها با استفاده از مقادیر از پیش تعیین‌شده مشخص می‌شود برای مثال

مقدار آستانه‌ای اول باید شامل ۳۳٪ از نمونه‌ها و مقدار آستانه‌ای دوم ۶۶٪ از نمونه‌ها را در برگیرد. این مسئله در ویژگی‌های بازه زمانی و تناوبی نیز رعایت شده است.

به‌دلیل ماهیت پویای بات‌نت‌ها مقادیر آستانه‌ای ایستا نمی‌تواند به‌اندازه کافی کارآمد باشد و این امر سبب می‌شود بات‌نت‌ها رفتار متغیری از خود داشته باشند. بنابراین بهترین راه برای مواجه با پویایی رفتار در بات- نتها استفاده از مقادیر آستانه‌ای پویا است.

در حالت پویا برخلاف روش مقداردهی ایستا که مقادیر آستانه‌ای برای همه بات‌نت‌ها یکسان و ثابت است، مقادیر آستانه‌ای مطابق با هر بات متفاوت و در فاز آموزش به چارچوب پیشنهادی آموزش داده خواهد شد.

در این روش با استفاده از داده آموزشی و بررسی هر کدام از ویژگی‌ها و همچنین دسته‌بندی آن‌ها در چارک‌ها و مشخص کردن مقادیر مرزی[1] موجود اقدام به تنظیم مقادیر آستانه‌ای مخصوص آن بات خواهد شد. این راهکار کاربرد بهتری دارد و می‌تواند به‌صورت بهینه‌تری مورداستفاده قرار گیرد.

۳-۸- الگو کردن و استخراج الگوها

در این مرحله اطلاعات به‌دست‌آمده از دسته اطلاعات ورودی به الگوهایی قابل‌فهم برای بررسی تبدیل می‌شود و الگوهایی برای فهم و آموزش سیستم ایجاد می‌شوند. در واقع در این با استفاده از جدول (۲) و مقادیر مشخص شده برای ویژگی‌های خارج جریانی همچون تناوبی بودن، برای هر یک از اتصال‌ها الگویی ایجاد می‌شود یا به‌نوعی می‌توان گفت با استفاده از جدول (۲) می‌توان کدگذاری انجام داد و با استفاده از آن هر یک از اتصال‌ها را به شکل یک رابطه کدی بیان کرد که در ادامه بتوان راحت‌تر آن را تحلیل نمود.

جدول (۲): جدول کدگذاری اتصال‌ها

اندازه	کوچک				متوسط				بزرگ				بسیار بزرگ			
بازه زمانی	کوتاه	متوسط	طولانی	بسیار طولانی	کوتاه	متوسط	طولانی	بسیار طولانی	کوتاه	متوسط	طولانی	بسیار طولانی	کوتاه	متوسط	طولانی	بسیار طولانی
تناوبی بودن																
قویاً تناوبی	a	b	c	j	d	e	f	k	g	h	i	l	o	m	n	p
تناوبی ضعیف	A	B	C	J	D	E	F	K	G	H	I	L	O	M	N	P
غیر تناوبی ضعیف	r	s	t	q	u	v	w	/	x	y	z	?	:	-	[	‘
غیر تناوبی قوی	R	S	T	Q	U	V	W	\	X	Y	Z	~	;	=	]	“
عدم اطلاعات کافی	1	2	3	!	4	5	6	@	7	8	9	#	$	%	^	&

همان‌طور که می‌توان مشاهده نمود جدول (۲) شامل سه ویژگی اندازه، بازه زمانی و تناوبی بودن هست. در سطر اول جدول اندازه‌ها بیان‌شده است و در سطر دوم بازه زمان مشخص‌شده است که مابین کوتاه، متوسط، طولانی و بسیار طولانی هست و در ستون اول از سمت راست نیز مقادیر مختلف تناوبی بودن مشخص‌شده است؛ بنابراین در اینجا با توجه به جدول کد a به معنای اندازه کوچک، بازه زمانی کوتاه و قویاً تناوبی[1] هست.

برای مثال یک جریان ورودی که همان اتصال هست که پیش‌تر توضیح داده شد در مورد آن به‌صورت زیر الگو می‌شود.

11Rrrrrarrrsrrarsrrrrrrrrrrarrarrarrrrarrarrarrrrrr

همان‌طور که مشاهده می‌شود این الگو دقیقاً منطبق بر جدول کدگذاری شده است و بدین شکل می‌توان روی جریان تحلیل بهتری را انجام داد و دانش بهتری را استخراج کرد و در ادامه نیز از آن به نحو بهتری استفاده نمود.

[1] Outlier

۳-۹- نحوه آموزش سیستم در روش پیشنهادی

سیستم پیشنهادی با استفاده از مجموعه جریانات کد شده که مورد تحلیل قرارگرفته است با استفاده از دو مجموعه جدا از هم یعنی مجموعه جریانات سالم و مجموعه جریانات ناسالم مورد آموزش قرار می‌گیرد. به ازای هر یک از مجموعه جریانات سالم، کدگذاری صورت گرفته و جریان به‌صورت کدی استخراج می‌شود که به آن درصورتی‌که از مجموعه سالم باشد برچسب سالم زده می‌شود و درصورتی‌که از مجموعه ناسالم باشد برچسب ناسالم زده می‌شود و در این حالت پایگاه داده‌ای ایجاد می‌شود که در حین آموزش رفتارهای مختلف موردبررسی قرار می‌گیرد و برای آن پایگاه داده‌ای از جریانات کدگذاری شده برای موارد سالم و مخرب به دست می‌آید.

۳-۱۰- بررسی جریانات جدید

برای اینکه بتوان یک مجموعه جریان جدید را پیش‌بینی نمود در ابتدا باید تمامی مراحل قبل روی آن صورت گیرد تا کدگذاری برای این جریان جدید به دست آید و در ادامه با استفاده از زنجیره مارکوف مقدار شباهت محاسبه می‌شود زیرا زنجیره مارکوف قادر است تا زیرمجموعه‌ای از یک مجموعه را موردبررسی قرار دهد و میزان تشابه را محاسبه نماید.

زنجیره مارکوف دوحالته، پایه‌ای‌ترین الگو از زنجیره مارکوف است. که در آن زنجیره مارکوف تنها دارای دو حالت خواهد بود. جهت تشریح زنجیره مارکوف معمولاً از این الگو استفاده می‌شود.

در اینجا از ماتریس تصادفی استفاده می‌شود و هر مقداری که در ماتریس تصادفی ذخیره می‌شود یک مقدار غیر منفی و بین صفر تا یک است و نشان‌دهنده بردار احتمال خواهد بود. به‌بیان‌دیگر گذار از یک حالت به حالت دیگر بر اساس یک احتمال اتفاق می‌افتد که آن احتمال در مکان متناظر در جدول قرار می‌گیرد. به‌طورکلی و فارغ از تعداد حالاتی که در یک زنجیره مارکوف وجود دارد سه نوع ماتریس احتمال برای هر زنجیره مارکوف وجود دارد:

- ماتریس احتمال راست: در این نوع ماتریس مجموع احتمالات هر سطر برابر ۱ خواهد شد
- ماتریس احتمال چپ: در این نوع ماتریس مجموع احتمالات هر ستون برابر ۱ خواهد شد.
- ماتریس احتمال دوطرفه: مجموع احتمالات در این ماتریس هم از ستون و هم از سمت سطر برابر ۱ خواهد شد.

می‌بایست در راهکار پیشنهادی برای تمامی رفتارهای موجود در پایگاه داده که از قبل در حین آموزش ایجاد کردیم، بررسی با استفاده از زنجیره مارکوف را انجام دهیم. به‌طورکلی زنجیره مارکوف میزان تشابه دو رشته را مشخص می‌کند و ما زمانی که از یک آستانه حدود ۷۰ درصد تشابه بیشتری را با یکی از رفتارهای ذخیره‌شده در پایگاه داده موردبررسی قرار می‌دهیم و درصورتی‌که به میزان تشابه بیش از آستانه برسیم آنگاه در همین نقطه بررسی قطع‌شده و تشخیص صورت گرفته است و در این حالت با توجه به برچسب رفتار ذخیره‌شده در پایگاه داده مشخص می‌کنیم که از نوع سالم است و یا نه. در این قسمت ۷۰ درصد در نظر گرفته‌شده است چراکه این مقدار یک مقدار دلخواه و اختیاری است و با توجه به مقالات مختلفی که خوانده‌شده است و در نظر گرفتن این مقالات بر عدد ۷۰ درصد در این قسمت نیز همین مقدار در نظر گرفته‌شده است.

فرض کنید دو رفتار به عنوان نمایندگی دو حالت از یک بات داریم که با رنگ های سیاه (حالت B) و سفید (حالت W) مشخص می شوند. فرض کنید رشته هایی داریم که می توانیم به هر یک از این حالت ها اختصاص دهیم. در صورتی که رشته رفتار ویا حالت سفید قرار بگیرد با W و در صورتیکه جز رفتار و یا حالت سیاه قرار گیرد با B نشان می دهیم. فرض کنید ۱۷ رشته داریم و به ترتیب مورد بررسی قرار می دهیم بنابراین یکی از حاالتی که اتفاق می افتد به‌صورت زیر خواهد بود.

WWBBWBWBBWBWBBBWW

احتمال گذار از حالت W به B را با P[W|B]نشان می دهیم. جهت محاسبه مقادیر ماتریس احتمال به طور کلی دو حالت داریم، یا در حالت W قرار داریم و یا در حالت B

در وضعیت W قرار داریم: در این حالت به طور کلی وضعیت بعدی یا B خواهد بود و یا W ،بنابراین خواهیم داشت WW و یا WB.

در وضعیت B قرار داریم در این حالت به طورکلی وضعیت بعدی یا B خواهد بود و یا W ،بنابراین خواهیم داشت BW و یا BB.

تعداد حالاتی که در وضعیت W قرار داریم را با رنگ دیگر مشخص کرده ایم تعداد این حاالت ۶ است:

WWBBWBWBBWBWBBBWW

بنابراین احتماالت $P_w\,[W\,|W]$, $P_w\,[W\,|B]$ به شکل زیر محاسبه می‌شوند:

$$P_w\,[\,W\,|B\,] = \frac{number\ of\ WB\ state}{number\ of\ W\ state} \qquad (۴)$$

$$P_w\,[\,W\,|W\,] = \frac{number\ of\ WW\ state}{number\ of\ W\ state} \qquad (۵)$$

به همین ترتیب برای حالت B داریم:

$$P_w\ [\ B\ |B\ ] = \frac{number\ of\ BB\ state}{number\ of\ B\ state} \quad (۶)$$

$$P_w\ [\ B\ |W\ ] = \frac{number\ of\ BW\ state}{number\ of\ B\ state} \quad (۷)$$

در نهایت جدول احتمال برای مثال بیان شده با استفاده از فرمول های بیان شده عبارت است از:

	W	B
W	$\frac{1}{6} = 0.166$	$\frac{5}{6} = 0.833$
B	$\frac{5}{8} = 0.625$	$\frac{3}{8} = 0.375$

با استفاده از این احتمالات بدست آمده می توان دسته بندی را برای هر رشته ورودی مبتنی بر رفتار بات مشخص نمود و بدین شکل این روند را برای تمام ورودی ها انجام داد.

به‌دلیل اینکه زنجیره مارکوف آموزش‌دیده ممکن است با زنجیره مارکوف مشاهده‌شده در ترافیک از نظر طول متفاوت باشد و ممکن است در نمونه‌ای که در حال پایش هستیم حالتی وجود داشته باشد که در زنجیره آموزش داده شده وجود ندارد. در این حالات به ترتیب ممکن است احتمال اشتباه محاسبه و یا صفر شود، بنابراین حالت ناموجود و اضافه در محاسبه احتمال تولید زنجیره حالات اصلی در نظر گرفته نمی‌شود. این فن در روش ما این قابلیت را ایجاد می‌کند که بات-های چندریختی را که با ایجاد ترافیک‌های متفاوت سعی در گمراهی سامانه می‌کنند را شناسایی کند. برای مثال در دو مجموعه زیر می‌توان مشاهده نمود که رشته کد گذاری شده در پایگاه داده شامل 81aaabababababa هست و در حین بررسی رشته دوم یعنی:

81aaaabf0fabababababababafxzfxfx

می‌توان مشاهده نمود که این رشته دوم شامل رشته اول است ولی برای اینکه بتواند پنهان بماند سعی کرده است تا رفتار اصلی خود را تقسیم کند ولی در زنجیره مارکوف با توجه به ماهیت عدم توجه به گذشته به‌راحتی قادر است تا روی این نوع از رشته‌ها تشابه را مشخص نماید.

Train Chain: 81aaabababababa
Monitor Chain: 81aaaabf0fabababababababafxzfxfx

۴- ارزیابی روش پیشنهادی

پیاده‌سازی در برنامه Visual Studio 2017 [۳۳] و با زبان برنامه‌نویسی C# صورت گرفته است. در این پیاده‌سازی از کتابخانه‌های وکا[1] [۳۴] و ZedGraph استفاده‌شده است. معیارهایی نیز در این تحقیق برای مقایسه و بررسی استفاده‌شده‌اند که در عبارت‌اند از:

- معیار صحیح مثبت (TP[2]): درصورتی‌که جوابی که از پیش‌بینی به دست می‌آید p باشد و مقدار واقعی نیز p باشد آنگاه ۱ واحد به TP اضافه می‌شود بنابراین در مجموعه داده‌هایی که برای آزمون استفاده می‌شود این مقدار جمع‌آوری می‌شود.
- معیار صحیح منفی (TN[3]): درصورتی که جوابی که از پیش‌بینی به دست می‌آید n باشد و مقدار واقعی نیز n باشد آنگاه ۱ واحد به TN اضافه می‌شود بنابراین در مجموعه داده‌هایی که برای آزمون استفاده می‌شود این مقدار جمع‌آوری می‌شود. تفاوت این قسمت با TP در این هست که در TP جواب مثبت و در اینجا جواب‌های منفی را شمارش می‌شود.
- معیار غلط مثبت (FP[4]): درصورتی‌که جوابی که از پیش‌بینی به دست می‌آید p باشد و مقدار واقعی n باشد آنگاه ۱ واحد به FP اضافه می‌شود بنابراین در مجموعه داده‌هایی که برای آزمون استفاده می‌شود این مقدار جمع‌آوری می‌شود. یعنی به‌طورکلی جوابی که از پیش‌بینی به‌دست می‌آید متفاوت از مقدار واقعی باشد.
- معیار غلط منفی (FN[5]): درصورتی‌که جوابی که از پیش‌بینی به دست می‌آید n باشد و مقدار واقعی p باشد آنگاه ۱ واحد به FN اضافه می‌شود بنابراین در مجموعه داده‌هایی که برای آزمون استفاده می‌شود این مقدار جمع‌آوری می‌شود. یعنی به‌طورکلی جوابی که از پیش‌بینی به دست می‌آید متفاوت از مقدار واقعی باشد. تفاوت این قسمت با قسمت قبل در این هست که در قسمت قبل جواب‌های منفی که به‌اشتباه مثبت تلقی می‌شدند، در نظر گرفته‌شده است و در اینجا جواب‌های مثبتی که به‌اشتباه منفی تلقی شده‌اند در نظر گرفته می‌شوند.

محیطی که پیاده‌سازی در آن صورت گرفت و نتایج ارزیابی شد دارای سخت‌افزاری با مشخصات زیر هست:

- حافظه ۱۶ گیگابایت
- پردازنده ۷ هسته‌ای
- کارت گرافیک ۲ گیگابایت
- حافظه ذخیره‌سازی ۲۵۶ گیگابایت درایو حالت‌جامد[6]

۴-۱- داده‌های مورداستفاده

در این پیاده‌سازی جریانات ورودی در ابتدا با استفاده از راهکار پیشنهادی جمع‌آوری و با استفاده از تحلیل روی جریان، مجموعه داده‌ای برای آن ایجاد شد که در ادامه به‌عنوان ورودی به سیستم

[1] weka
[2] True Positive
[3] True Negative
[4] False Positive
[5] False Negative
[6] Solid-state drive(SSD)

داده می‌شود. در این دیتاست به‌صورت نظارت‌شده عمل شده است یعنی حملات و غیر حملات و همین‌طور نوع زیر دسته حمله به‌طور دقیق مشخص‌شده است و در این قسمت هدف آموزش سیستم و در ادامه استفاده از راهکار cross k fold برای بررسی راهکار پیشنهادی در مقایسه با دیگر روش‌ها هست. دیتاست به‌دست‌آمده شامل ۵۵۹۴۰ رکورد هست و به‌صورت فایل csv ذخیره شدند که یک فرمت کاملاً استاندارد برای ذخیره‌سازی رکوردها با حجم بالا هست. این داده‌ها برچسب‌گذاری شده هستند بنابراین کار از نوع نظارت‌شده هست. در اینجا مقدار k برای روش استفاده از روش cross k fold برابر ۱۰ قرار داده‌شده است که این مقدار یک مقدار متعارف در زمینه داده‌کاوی هست. در راهکارهای گذشته از روش جداسازی[1] استفاده می‌شد که معمولاً ۶۶ درصد داده‌ها از ابتدا به‌عنوان داده‌های آموزش و مابقی به‌عنوان داده‌های آزمون در نظر گرفته می‌شدند ولی در این حالت ممکن است در مجموعه آزمون رفتارهایی دیده شود که اصلاً در مجموعه آموزش وجود نداشته است ازاین‌رو cross k fold دارای محبوبیت بیشتری هست زیرا تمامی قسمت‌های مختلف مجموعه داده به‌عنوان آموزش و آزمون قرار می‌گیرند و این درنهایت نتایج میانگین مراحل مختلف بررسی این روش قرار می‌گیرد و به‌صورت تجربی اثبات‌شده است که مقدار k=10 بهترین مقدار برای k هست که در این تحقیق نیز k برابر با ۱۰ در نظر گرفته می‌شود.

۴-۲- تحلیل نتایج

در این بخش روش پیشنهادی با روش‌های [۹]، [۲۷] و [۳۰] مورد مقایسه قرار می‌گیرد. برای مقایسه، تمامی روش‌های بیان‌شده در [۹]، [۲۷] و [۳۰] پیاده‌سازی و با توجه به پیاده‌سازی صورت گرفته نتایج به‌دست آمدند که بدین شکل به توان دقیق‌تر بررسی‌ها را انجام داد.

داده‌ها در ابتدا توسط برنامه به فرمت مناسب برای تحلیل قرار می‌گیرد یا به عبارتی پیش‌پردازش ابتدایی صورت می‌گیرد، فایلی با فرمت ARFF ایجاد می‌شود که ساختاری مناسب و استاندارد برای تحلیل هست.

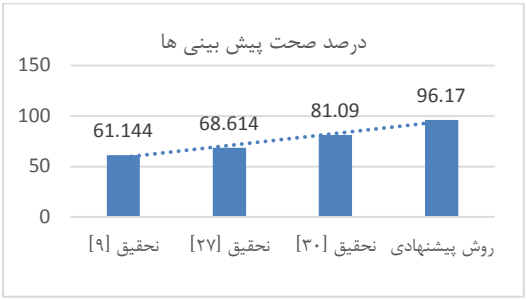
جدول (۳): نسبت درصد صحت پیش‌بینی‌ها و خطای پیش‌بینی‌ها

	درصد صحت پیش‌بینی‌ها	درصد خطای پیش‌بینی‌ها
روش تحقیق [۹]	٪۶۱,۱۴۴	٪۳۸,۸۵۵
روش تحقیق [۲۷]	٪۶۸,۶۱۴	٪۳۱,۳۸۵
روش تحقیق [۳۰]	٪۸۱,۰۹۰	٪۱۸,۹۰۹
روش پیشنهادی	٪۹۶,۱۷۰	٪۳,۸۲۹

با توجه به نتایج دریافتی از جدول (۳) می‌توان به‌وضوح مشاهده نمود که الگوریتم پیشنهادی با ٪۹۶,۱۷۰ دقت دارای بالاترین دقت صحت و با ٪۳,۸۲۹ اشتباه دارای کمترین میزان اشتباه هست. می‌توان در جدول (۳) نسبت بهتری را مشاهده نمود. کاملاً مشخص است که الگوریتم پیشنهادی از باقی روش‌ها بسیار بهتر عمل کرده است و این نسبت بهبود نسبت به دیگر راهکارها کاملاً مشهود هست و می‌توان گفت دلیل اصلی این قضیه استفاده از مارکوف در الگوریتم پیشنهادی هست.

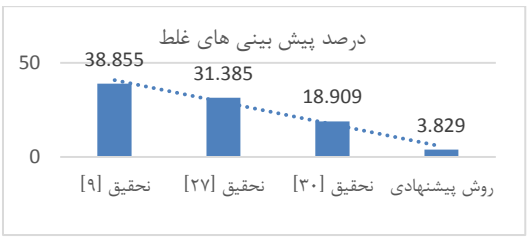
با توجه به این دقت‌های به‌دست‌آمده کاملاً قابل مشاهده هست که روش پیشنهادی از دیگر روش‌ها بهتر عمل کرده است و دراین‌بین الگوریتم تحقیق [۳۰] از الگوریتم‌های ارائه‌شده در تحقیقات [۹] و [۲۷] بهتر عمل کرده است و همچنین الگوریتم ارائه‌شده در تحقیق [۲۷] از راهکار ارائه‌شده در تحقیق [۹] بهتر هست. در این میان با توجه به نتایج دریافتی نتیجه مشهود این است که روش پیشنهادی قادر بوده است تا با عملکرد بسیار بالا و دقت بالایی حملات را درست تشخیص دهد و نشان از مفید بودن این راهکار در مقابل دیگر روش‌های مورد مقایسه هست.

همان‌طور که می‌توان از جدول (۳) دریافت روش پیشنهادی دارای دقت تشخیص صحیح بیشتری نسبت به دیگر الگوریتم‌های مورد بررسی یعنی [۹]، ۲۷ و ۳۰] هست. این بدین دلیل هست که ما در روش پیش‌بینی خود تنها مواردی از داده‌های آزمون را در نظر گرفتیم که تأثیر بیشتری را در نتیجه خروجی داشتند و در نتیجه داده‌هایی که در خروجی تأثیر نداشتند استفاده‌نشده است و بدین شکل زمان تحلیل کاهش‌یافته است.



شکل (۵): درصد پیش‌بینی صحیح در میان داده‌های آزمون برای الگوریتم پیشنهادی و دیگر الگوریتم‌های موردبررسی

در نمودار ارائه‌شده در شکل (۶) می‌توان درصد تشخیص غلط را مشاهده نمود.



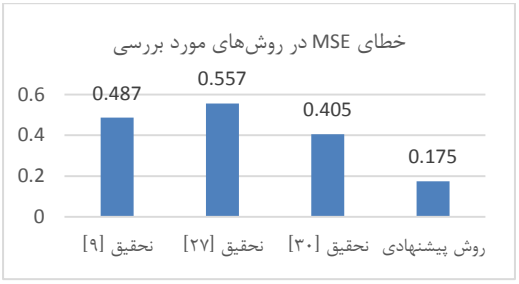
شکل (۶): درصد پیش‌بینی ناصحیح (غلط) در میان داده‌های آزمون برای الگوریتم پیشنهادی و دیگر الگوریتم‌های موردبررسی.

[1] split

در شکل (۷)، خطای میانگین مربعات (MSE[1])[۳۵] برای روش پیشنهادی و دیگر روش‌های مورد مقایسه محاسبه‌شده است. این معیار میانگین خطای مربعات را با استفاده از رابطه (۴) به دست می‌آورد و با استفاده از این معیار می‌توان به‌طور دقیق پیش‌بینی را سنجید و میزان خطای پیش‌بینی را محاسبه نمود. در این پیاده‌سازی از این رابطه برای محاسبه خطا پیش‌بینی استفاده می‌شود که مشاهده شود تا چه اندازه روشی که برای پیش‌بینی باتنت‌ها معرفی کردیم قادر است تا به‌درستی پیش‌بینی‌ها را انجام دهد.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(r_i - p_i)^2 \qquad [۳۵]\ (۴)$$

که در آن N تعداد کل داده‌های آزمایش، r مقدار واقعی و p مقدار پیش‌بینی‌شده است.



شکل (۷): میزان معیار MSE در میان الگوریتم پیشنهادی و دیگر الگوریتم‌های مشابه موردبررسی.

۵- جمع‌بندی

در این پژوهش راهکاری برای ارزیابی و پیش‌بینی حملات ارائه شد و مشاهده گردید که روش ارائه‌شده دارای عملکرد مناسبی بوده و بهبود نسبتاً بالایی را نسبت به الگوریتم‌های مقالات [۹]، [۲۷] و [۳۰] دارا هست. نوآوری اصلی در این تحقیق را می‌توان شامل دو بخش دانست: ۱-استفاده از راهکاری برای استخراج قسمت‌هایی از جریان‌ها که امکان به‌اشتباه افتادن کمتر شود. ۲- استفاده از راهکاری که نیازمند به خاطر سپردن گذشته نیست و بدین شکل می‌توان به شکل مؤثرتر و سربار محاسباتی کمتری، باتنت‌ها را شناسایی نمود. در گذشته هیچ مقاله ای به استفاده از زنجیره مارکوف بدین هدف و شکل که در این مقاله مورد استفاده قرار گرفته است، نپرداخته است بنابراین کار انجام شده در این مقاله کاری کاملا جدید است که دارای قدرت تشخیص خوبی نیز می باشد.

برای تقسیم داده آموزش و آزمون نیز از روش Cross K-Fold استفاده شد زیرا در این روش بهترین حالت بررسی صورت می‌گیرد و تمامی قسمت‌های دادها مشاهده می‌شود درحالی‌که استفاده از راهکارهایی همچون Split تمامی قسمت‌های داده‌ها مشاهده نمی‌شود. برای مثال ممکن است یکسری از حالات تنها در قسمت آزمون قرار بگیرد و در این حالت چون سیستم در قسمت آموزش این حالات را ندیده است دقت تشخیص پایین می‌آید بنابراین باید بهترین حالت آزمون و بررسی صورت بگیرد که چون در حالت k-fold تعداد به بخش‌های مستقلی تقسیم و در هر مرحله قسمتی از آن برای آزمون قرار می‌گیرد تمامی قسمت‌ها بررسی می‌شوند و بهترین حالت بررسی صورت می‌گیرد. در علم داده‌کاوی داده بسیار دارای اهمیت بالایی هست زیرا این داده‌ها هستند که باعث ایجاد علم و پیش‌بینی می‌شوند در این تحقیق از مجموعه داده‌ای با حدود ۵۵۹۴۰ رکورد استفاده‌شده است که این دیتاست در نتیجه تحلیل جریانات ورودی ایجاد شده است. در اینجا داده‌ها در ابتدا مورد پیش‌پردازش قرار گرفت و انتقالاتی نیز روی داده‌ها صورت گرفت تا اینکه داده‌ها به داده‌های ورودی موردنیاز الگوریتم پیشنهادی تبدیل شوند. نتایج به‌دست‌آمده کاملاً بهبود روش پیشنهادی را نشان می‌دهد.

پیشنهاد زیر را می‌توان در راستای این تحقیق ارائه نمود:

- می‌توان به‌جای استفاده از زنجیره مارکوف در اینجا مجموعه داده‌ای ایجاد نمود که شامل امضا نباشد و در انتها برای بررسی و ساخت الگو از درخت تصمیم استفاده نمود که در نتیجه آن می‌توان، تصمیمات بسیار سودمندی را نیز استخراج نمود که در چه حالت‌هایی ممکن است حمله باشد و یا نباشد ولی یکی از مشکلاتی که در استفاده از این روش پیش می‌آید، حافظه است یعنی می‌بایست حافظه بسیار بالایی را برای استفاده از این روش در اختیار داشت که اگر جریان‌ها بسیار بزرگ باشند عملاً استفاده از روش درختی امکان‌پذیر نیست ولی اگر بتوان راهکاری برای کاهش حافظه در الگوریتم درختی ارائه نمود می‌توان گفت روشی بسیار قوی‌تر از راهکار این مقاله به دست می‌آید که قادر است نتایج بسیار سودمندتری را نیز در اختیار قرار دهد.

۶- مراجع

[1] M. Khanjani, "Software Blurring by Multi-Yarn Petri Nets", 20th Annual National Conference of the Iranian Computer Association, 2015. (In Persian)

[2] P. Miller, "Hybrid Analysis and Control of Malware," Computer Sciences Department, 2017.

[3] J. B. A. Z. Bosnić, "Extending applications using an advanced approach to dll injection and api hooking," Practice and Experience Journal, vol. 40, pp. 567-584, 2010.

[1] Mean Squared Error

[4] M. Vaziri, “Finding Bugs with a Constraint Solver,” MIT Laboratory for Computer Science, Massachusetts, 2018.

[5] https://www.hex-rays.com/products/ida/, Hex-Rays. IDA Pro, Last access: March 18, 2016.

[6] A. Tan, M. Steinbach, and V. Kumar, “Introduction to Data Mining,” Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, First Edition, 2015.

[7] C. E, J. Faster and M. Degory, “The zombie roundup: understanding, detecting, and disrupting botnets,” SRUTI, 2005.

[8] C. Hester, L. Helia, and K. Hour, “BotGAD: detecting botnets by capturing group activities in network traffic,” In Proceedings of the Fourth International ICST Conference on Communication System Software and Middleware, 2009.

[9] Gu G. R. Perdisci, J. Zhang, and W. Lee, “BotMiner: Clustring Analysis of NetworkTraffic for Protocol- and Structure- Independent Botnet Detection,” in Proceedings of the 17th USENIX Security Symposium, Sanjose, CA, USA. 2018

[10] H. Duc, T. Yan, G. Eidenbenz, S. Ngo, and H. Qeue, “Botnets,” IEEE dependable systems and networks conference, pp. 297-306, 2019.

[11] K. Kenji and R. Larry, “The Feature Selection Problem: Traditional Methods and a New Algorithm,” AAAI-92 Proceedings, 2016.

[12] O. D. Inc., “The Role of DNS in Botnet Command and Control,” 2012.

[13] M. Antonakakis, C. Elisan, D. Dagon, G. Ollmann, and E. W. Damballa, “The Command Structure of the Aurora Botnet,” 2010.

[14] Y. Zeng, X. Hu, and G. Shin, “Detection of Botnets Using Combined Host and Network-Level Information, ” IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), pp. 291-300, 2017.

[15] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, “Using Machine Learning Techniques to Identify Botnet Traffic, ” IEEE Internetwork Research Department BBN Technologies, proceeding 31th IEEE conference, pp. 967–974, 2016.

[16] Z. Foladi, H. Hani, Y. Farjami, and J. Rezaei, “Discovery of botnets based on network traffic behavior, the first national conference on new approaches in computer engineering and information retrieval, Rudsar, ” Islamic Azad University of Rudsar and Amlash Branch, 2013. (In Persian)

[17] Y. Shang, “Botnet Detection with Hybrid Analysis on Flow Based and Graph Based Features of Network Traffic, ” International Conference on Cloud Computing and Security, pp. 612-621, 2018.

[18] E. Stinson, J. Mitchell, “Characterizing bots’ remote control behavior,” In Detection of Intrusions & Malware, and Vulnerability Assessment, 2007.

[19] Z. Chi, Z. Jin, and Ch. Zheng, “Botnet detection based on behavior analytics, ” pp. 612-621, 15. 03. 2018.

[20] W. T. Strayer, R. Walsh, C. Livadas, and D. Lapsley, “Detecting botnets with tight command and control,” In Local Computer Networks, Proceedings 31st IEEE Conference, 2016.

[21] J. Goebel and T. Holz, “Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation,” Hotbots, 2017.

[22] B. Qi, J. Jiang, Z. Shi, R. Mao, and Q. Wang, “Detecting DGA-Based Botnet Using Two-Stage Anomaly Detection, ” In IEEE, New York, NY, USA, 2018.

[23] P. G. Efthimion and S. Payne, “Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots, ” SMU Data Science Review, vol. 1, p. 52, 2018.

[24] A. Karasaridis, B. Rexroad, and D. Hoein, “Wide-Scale Botnet Detection and Characterization,” Workshop on Hot Topics in Understanding Botnets, 2017.

[25] T. Cochran and J. Cannady, “Not so fast flux networks for concealing scam servers,” in Risks and Security of Internet and Systems (CRiSIS), 2010.

[26] A. Maroussi, I. Zabah, and H. Khabaz Atai, “Network intrusion detection using a combination of artificial neural networks,” In a hierarchical manner, Electronic and Cyber Defense, vol. 8, no. 1, pp. 89-99, 2020. (In Persian)

[27] K. Wang, C. Huang, S. Lin, and Y. Lin, “A fuzzy pattern-based filtering algorithm for botnet detection,” Computer Networks, vol. 55, no. 15, pp. 3275–3286, 2011.

[28] B. A. AlAhmadi and I. Martinovic, “Malware family classification using network flow sequence behavior, ” in APWG Symposium on Electronic Crime Research, San Diego, CA, USA, 2018.

[29] K. Shoshian, A. Rashidi, A. Jabbar, and M. Dehghani, “Transport of ambiguous cyber model based on alternative attack,” Electronic and Cyber Defense, vol. 8, no. 1, pp. 67-77, 2020. (In Persian)

[30] V. I. Ghafir, “A System for Real Time Botnet Command and Control Traffic Detection, ” Cyber-Threats and Countermeasures in the Healthcare Sector, vol. 6, pp. 38947 - 38958, 2018.

[31] S. Ledesma, G. Cerda, G. Avina, D. Hernandez, M. Torres, A. Gelbukh, and E. F. Morales, “Feature Selection Using Artificial Neural Networks,” MICAI 2008, LNAI 5317, pp. 351–359, 2008.

[32] Y. Xiaocong, D. Xiaomei, Y. Ge, Q. Yuhai, and Y. Dejun, “Data-Adaptive Clustering Analysis for Online Botnet Detection,” In Proceedingd of the 3th IEEE International Joint Conference on Computational Science and Optimization, Anhui, China, 2016.

[33] “Microsoft Visual Studio 2015 Language Pack,” Microsoft.com. Microsoft, 2019.

[34] https://www.cs.waikato.ac.nz/ml/weka/, 2020.

[35] D. Wackerly, W. Mendenhall, and R. Scheaffer, “Mathematical Statistics with Applications (7 Ed.),” Belmont, CA, USA: Thomson Higher Education, ISBN 0-495-38508-5, 2008.