نشریه علمی "پدافند الکترونیکی و سایبری"
سال هشتم، شماره ۴، زمستان ۱۳۹۹، ص ۱۱۸- ۱۰۷

علمی- پژوهشی

بهبود روش شناسایی باج‌افزارها با استفاده از ویژگی‌های توابع سیستمی

حمیدرضا جواهری[۱]، حمید اکبری[۲*]، احسان اله شقاقی[۳]

۱- کارشناسی ارشد، ۲- استادیار، ۳- کارشناسی ارشد، دانشکده سایبرالکترونیک دانشگاه جامع امام حسین (ع)

(دریافت: ۱۳۹۸/۰۹/۱۹، پذیرش: ۱۳۹۸/۱۱/۱۲)

چکیده

در سال‌های اخیر گرایش حملات سایبری مبتنی بر باج افزارها به‌شدت افزایش‌یافته است. یکی از روش‌های پدافندی، شناسایی رفتاری باج‌افزارها به‌وسیله توابع سیستمی است. با مطالعه و بررسی پژوهش‌های این حوزه دریافتیم پژوهش‌های مذکور در نرخ دقت و سرعت تشخیص باج‌افزارها بهینه نمی‌باشد. به‌دلیل اینکه جامعه آماری نمونه باج‌افزارهای مجموعه‌داده‌های مورد آزمایش و ارزیابی در این پژوهش‌ها محدود بوده و همه خانواده‌های باج‌افزاری را پوشش نمی‌دهد، لذا میزان نرخ‌های تشخیص ارائه‌شده برای شناسایی تعداد بالای باج‌افزارها دارای کاستی‌هایی چون پایین بودن نرخ دقت تشخیص، نرخ بالای مثبت کاذب و حتی بالا بودن نرخ عدم‌تشخیص هستند. از دیگر کاستی پژوهش‌های مذکور غفلت از تأثیر نرخ سرعت در تشخیص باج‌افزارها است. عدم رفع کاستی‌های مذکور در زمان پیاده‌سازی این‌گونه روش‌های شناسایی، موجب متحمل شدن هزینه‌های زمانی و مادی زیادی و نیز موجب کندی سیستم شناسایی و عدم دستیابی به خروجی صحیح و واقعی خواهد شد. لذا در این پژوهش ابتدا اقدام به تولید مجموعه‌داده غنی شامل انواع خانواده باج‌افزارها و در نسخه‌های مختلف شده است. در ادامه با انجام آزمون‌هایی طی ۴ مرحله روی مجموعه‌داده اولیه با ۱۲۶ ویژگی و برگزیدن الگوریتم انتخاب ویژگی مناسب، اقدام به بهینه‌سازی آن شده است. در نتیجه مجموعه‌داده‌ای بهینه با ۶۷ ویژگی بدون کاهش نرخ دقت تشخیص به‌دست آمده است. سپس به‌وسیله این مجموعه‌داده بهینه و به اصطلاح سبک اقدام به اخذ بهترین مدل دسته‌بندی برای تشخیص کرده، لذا به‌وسیله الگوریتم دسته‌بندی جنگل تصادفی (با استفاده از روش مقابله‌ای ۱۰ بخشی) موفق به شناسایی باج‌افزارها با نرخ دقت بهینه ۹۵۶۷/۱۱٪ در مدت زمان ۰/۲۱ ثانیه، نرخ مثبت کاذب ۰/۰۴۷ و نرخ مثبت صحیح ۰/۹۵۱ شده‌ایم.

کلیدواژه‌ها: باج‌افزار، شناسایی رفتاری باج‌افزارها، انتخاب ویژگی‌های باج‌افزارها، باج‌افزارهای رمزنگار، توابع سیستمی، نرخ دقت و سرعت تشخیص باج‌افزارها، دسته‌بندی باج‌افزارها.

۱- مقدمه

در دنیای امروز فناوری اطلاعات، تحولات زیادی وجود دارد. این تحولات در حوزه سایبری و امنیت آن اتفاق می‌افتد. در این حوزه روزانه ابزارهای بداندیش[۱] زیادی توسعه و تولید می‌شوند. در مقابل متخصصان امنیتی این حوزه سعی در شناسایی و جلوگیری از این‌گونه فعالیت‌ها دارند. طی گزارش از سوی اتحادیه اروپا باج‌افزارها با توجه به بهره‌مندی از کانال ارتباط فرماندهی و کنترل با سرورهای حمله‌گر به‌عنوان باتنت دسته‌بندی شده‌اند [۱]. باج‌افزارها معمولاً با قرنطینه کردن سیستم قربانی یا اطلاعات درون آن به‌وسیله رمزنگاری فایل‌ها، اطلاعات قربانی و یا قفل‌کردن سیستم برای رسیدن به اهداف خود تلاش می‌کند [۲].

ابعاد حوزه جرائم سایبری روزانه در حال رشد است و به‌طور طبیعی برنامه‌های بداندیش نیز به‌عنوان یکی از زمینه‌های این حوزه موردتوجه بوده است [۲]. طبق گزارش شرکت امنیتی سیمانتک حجم گونه‌های مختلف باج‌افزار نسبت به سال ۲۰۱۵ در سال ۲۰۱۷[۲] حدود ۴۶٪ افزایش داشته است [۳]. دیگر زمینه بسیار مهم قابل‌ذکر رشد روزافزون صنعت یا تجارت پول‌های الکترونیکی است که باعث رشد تولید خانواده‌ای از بدافزار به نام باج‌افزار شده است [۴]. باج‌افزارها دسته‌ای از بدافزارها هست که با استفاده از مهارت‌های بداندیش و مخرب دسترسی کاربران به اطلاعات سیستم‌هایشان جلوگیری می‌کند [۵].

طبق گزارش تهدیدات سال ۲۰۱۷ اتحادیه اروپا، حملات باج‌افزاری نسبت به دیگر تهدیدات جرائم سایبری رشد بسزایی داشته است [۲]. در ۶ ماه اول سال ۲۰۱۷ شاهد حملات مبتنی بر باج‌افزار با مقیاس دیده نشده نسبت به قبل بوده است. در این حملات شاهد ظهور باج‌افزارهای خود تکثیر مانند WannaCry و Petya بوده است. باج‌افزارها و سارقان اطلاعات به‌عنوان تهدیدات

*رایانامه نویسنده مسئول: hamidakbari@ihu.ac.ir

[1] Malicious

اصلی بدافزاری در سال ۲۰۱۷ هستند که در سرتاسر اتحادیه اروپا با قوانین جدی مورد برخورد قرار می‌گیرند.

با توجه به مباحثی که به‌عنوان مقدمه در این بخش ارائه گردید، لزوم پرداختن به بدافزارهایی چون باج‌افزارها بیش‌ازپیش مشخص گردید. مراحل شناسایی باج‌افزارها به دو صورت قابل‌طرح است. اول، شناسایی قبل از ورود و ساکن شدن باج‌افزار در سیستم قربانی که به‌نوعی روش‌های ایستا از قبیل شناسایی مبتنی بر امضاء، شناسایی مبتنی بر الگو یا ظروف عسل و دیگر روش‌های موجود در این زمینه مطرح است. درروش دوم شناسایی بر اساس تحلیل باج‌افزار و پس از مقیم شدن آن در سیستم قربانی صورت می‌گیرد [۶]. البته این روش‌ها هم‌زمان‌بر هستند و هم احتمال موفقیت در آن پایین‌تر است چرا که پس از نفوذ باج‌افزار و شروع فعالیت آن یافتن رد پا، شناسایی و کشف فایل‌های وابسته به آن و نیز جلوگیری از شیوع و سیر عملکردی آن دشوارتر از شناسایی به روش ایستا و قبل از مقیم شدن باج‌افزار در سیستم قربانی است. لذا ما در نظر داریم در این مقاله از روش تحلیل پویا یا رفتاری استفاده کرده تا قبل از مقیم‌شدن باج‌افزار در سیستم قربانی به شناسایی آن بپردازیم.

با توجه به مطالعات و بررسی‌های صورت گرفته در نظر داریم در این مقاله در زمینه شناسایی باج‌افزارها و در حوزه سیستم فایل‌ها و توابع فراخوانی شده، پژوهشی داشته باشیم. دلیل این انتخاب این است که باج‌افزارها در زمینه سیستم فایل‌ها فعالیت‌های بیشتر و خاص منظوره‌تری چون قفل‌کردن و یا رمزنگاری کردن، (که پیش‌تر بیان گردید) دارند، لذا شناسایی از این طریق مؤثرتر و با دقت بالاتری ممکن است. البته در این زمینه فعالیت‌هایی صورت گرفته است اما با مطالعه و بررسی دقیق‌تر این پژوهش‌ها مشخص گردید که کارهای مرتبط در این زمینه با کاستی‌هایی چون نرخ کم‌دقت و سرعت برخوردارند و کمتر به مسئله بهینه کردن روش‌های شناسایی باج‌افزارها پرداخته شده است. به‌طورمعمول تمام پژوهش‌های صورت گرفته در این عرصه روی حجم محدودی از باج‌افزارها (زیر ۵۰۰ نمونه) صورت گرفته است. اگر بتوانیم نرخ سرعت شناسایی در این پژوهش‌ها را بهینه و افزایش دهیم و یا نرخ دقت را بالاتر ببریم (حتی در حد خیلی محدود) برای پیاده‌سازی و کاربردی کردن این پژوهش‌ها در حجم بسیار بالا (مثلاً صدها میلیون باج‌افزار) اثر بسزایی خواهد داشت. همچنین با بهینه کردن این روش‌ها (مثلاً بهینه در مجموعه‌داده و ویژگی‌های آن) زمان پیاده‌سازی در زیرساخت‌های سخت‌افزاری و منابع سیستمی کمتر استفاده شده و در نتیجه هزینه مادی کمتری مورد نیاز است. این نکته یعنی بهینه‌کردن روش‌های موجود شناسایی باج‌افزارها خلاء قابل ملاحظه‌ای بود که در روش‌ها و پژوهش‌های موجود در این زمینه محسوس بوده و در این پژوهش سعی بر این است تا بتوان یک حالت بهینه در نرخ دقت و سرعت یکی از روش‌های موجود ایجاد نمود تا اثربخشی این روش‌ها در جهت شناسایی باج‌افزارها بیشتر گردد.

۲- پیشینه تحقیق

در این زمینه و در روش شناسایی باج‌افزارها مبتنی بر توابع سیستمی کارهای زیادی انجام گرفته است. لذا در ادامه به تشریح آن‌ها پرداخته‌ایم.

یو لون وان و همکارانش طرحی را ارائه داده‌اند که بر مبنای توسعه یک مدل تشخیص حملات بر مبنای داده‌های حجیم است [۷]. این طرح به پردازش، ترکیب و نشانه‌گذاری بسته‌های استخراج‌شده از رشته‌های شبکه‌ای پرداخته، سپس ویژگی‌های یک ترافیک کامل را استخراج کرده‌اند. در ادامه شش الگوریتم مربوط به انتخاب ویژگی با یکدیگر ترکیب شده تا به دقت بالاتری برای طبقه‌بندی ویژگی‌ها به‌دست آید. در نهایت مدل درخت تصمیم مبتنی بر یادگیری ماشین جهت بهبود عملکرد سیستم تشخیص نفوذ مورد استفاده قرار گرفته است.

در طرحی احمد الکوسیری و ماریان مازر، با استفاده از هانی‌پات‌ها به بررسی و سپس الگونگاری دسترسی‌های غیرمجاز گونه‌های باج‌افزار پرداخته‌اند [۸]. آن‌ها از طریق این الگوها و بر روی حملات به شناسایی باج‌افزارها اقدام کرده‌اند و در ادامه نسبت به هشداردهی موارد احصاءشده اقدام کرده‌اند.

نیکلای همپتون و همکارانش، به بررسی رفتاری باج‌افزارهای آلوده‌کننده سیستم‌عامل ویندوز پرداخته‌اند [۹]. سپس بر اساس نتایج حاصله و بررسی آن‌ها مشخص شده است که برخی از باج‌افزارها با میزان فرکانس خاصی و در بازه زمانی محدودی در قالب توالی خاصی از توابع اقدام به فراخوانی توابع مربوطه نسبت به نمونه‌های پاک کرده‌اند. در ادامه با الگو قرار دادن این ویژگی اقدام به شناسایی باج افزارها نموده‌اند.

کریشنان و سومن در مدلی به‌نام GURLS به شناسایی و دسته‌بندی باج‌افزارها پرداخته‌اند [۱۰]. ویژگی‌های استخراج‌شده در این طرح از تحلیل ایستا و نیز رشته‌های مورد استفاده باج‌افزار به‌دست آمده است. آن‌ها برای ایجاد این پروفایل و دسته‌بندی مربوطه از الگوریتم یادگیری RLS استفاده کرده‌اند و نیز برای استخراج ویژگی‌های فراخوانی‌های توابع و نیز رشته‌های در حال اجرا از جعبه شن کوکو استفاده می‌کنند و در نهایت از فرکانس حاصله از فراخوانی‌های توابع استخراج‌شده برای فرایند دسته‌بندی استفاده کرده‌اند.

ژی گو چن و همکارانش با استفاده از روش‌های داده‌کاوی به تشخیص پویای باج‌افزارهای شناخته و ناشناخته پرداخته‌اند [۱۱]. در این روش به مانیتور کردن رفتار باج‌افزار و استخراج گراف جریان فراخوانی توابع پرداخته‌اند. با استفاده از برنامه Api Monitor نسبت به رصد و استخراج جریان توابع فراخوانی‌شده توسط برنامه‌های مخرب و پاک و اخذ گراف توالی جریان آن اقدام شده است. در گام آخر با استفاده از الگوریتم‌های داده‌کاوی به مدل‌سازی تشخیص پرداخته تا به‌وسیله آن اقدام به قضاوت درباره پاک یا باج‌افزار بودن یک نمونه کنند.

یوکی تاکشی و همکارانش، روشی برای شناسایی باج‌افزارهای نوین بر مبنای الگوریتم یادگیری ماشین SVM ارائه شده است [۱۲]. در این طرح ابتدا فایل‌های واقعی باج‌افزار سطح وب‌سایت‌ها را جمع‌آوری کرده‌اند. سپس این باج‌افزارها در محیط جعبه‌شن به اجرا درآورده شده‌اند. فایل‌های مربوطه توسط جعبه شن مورد تحلیل پویا قرار گرفته‌اند. گزارش‌های حاصل از تحلیل پویای این باج‌افزارها از جعبه شن استخراج گردیده است. بر اساس ویژگی‌های خاص توابع فراخوانی‌شده توسط باج‌افزارها، فعالیت‌های مخرب و مشکوک شناسایی شده است و بر اساس ویژگی‌های استخراج‌شده بردار ویژگی‌ها تشکیل شده است. در ادامه به استانداردسازی بردار ویژگی‌ها برای همگام‌سازی فایل‌های گوناگون و متنوع شده است. سپس اقدام به مدل‌سازی نحوه تشخیص باج‌افزارها به‌وسیله الگوریتم SVM کرده‌اند. در نهایت این طرح به شناسایی باج‌افزارهای ناشناخته پرداخته است. پس از آزمایش‌های این طرح مشخص شد پس از مقایسه طرح ریک و همکارانش با این طرح نرخ دقت تشخیص به ترتیب ۹۴/۱۸ و ۹۷٪/۴۸ بوده است.

در نتیجه‌گیری این بخش باید بیان داشت که با توجه به پژوهش‌های مرتبط انجام‌شده در این حوزه و بررسی‌های صورت گرفته روی این پژوهش‌ها و نیز نتایج به‌دست‌آمده حاصل از آن، می‌توان کاستی‌های این پژوهش‌ها را در ۳ مورد بیان کرد:

- نبود یک مجموعه‌داده مستقل و همچنین غنی و جامع از انواع خانواده‌های باج‌افزار جهت حصول به خروجی دقیق و جامع‌تر. در بسیاری از این طرح‌ها نمونه‌های بسیاری از گونه‌های مهم و خطرناک باج‌افزاری، به‌دلیل شیوع کم آن در نظر گرفته نشده است.
- نرخ دقت تشخیص در اکثر طرح‌ها بالا بوده و قابل بهبود و افزایش یافتن است.

در هیچ یک از طرح‌ها و پژوهش‌های مرتبط عامل نرخ سرعت تشخیص به‌عنوان یک شاخص مهم در نظر گرفته نشده است. با در نظر داشتن این شاخص مهم در بهینه کردن روش تشخیص، تسریع در شناسایی باج‌افزارها حاصل می‌شود.

شایان ذکر است که طرح پیشنهادی مشروح در این مقاله در گام اول جامعیت کشف انواع گونه‌های باج‌افزار گسترده می‌شود و در گام دوم نرخ شناسایی باج‌افزارها هم از لحاظ دقت و هم‌زمان تشخیص بهینه خواهد شد.

۳- طرح پیشنهادی

در بخش شناسایی رفتاری باج‌افزار زمینه‌های شناسایی متعددی وجود دارد. می‌تواند شناسایی مبتنی بر ترافیک شبکه صورت پذیرد. از دیگر روش‌های شناسایی رفتاری استفاده از ظرف عسل است. در این مقاله در زمینه بررسی سیستم فایل‌ها و توابع فراخوانی شده توسط باج‌افزارها پژوهش شده است. دلیل این انتخاب این است که باج‌افزارها در زمینه سیستم فایل‌ها فعالیت‌های بیشتر و خاص منظوره‌تری چون عملیات قفل‌کردن سیستم و یا رمزنگاری دارند. این‌گونه بدافزارها برای اجرای خود و عملیاتی ساختن مأموریت خود در ویندوز نیاز به فراخوانی توابع سیستمی دارند و چون مجبور به این عمل هستند این خود شاخصی برای شناسایی باج‌افزارها می‌باشد. لذا شناسایی از این طریق، مؤثرتر و با دقت بالاتری ممکن است.

در این بخش ابتدا به تشریح نحوه تشکیل مجموعه‌داده که شامل جمع‌آوری فایل‌های نمونه باج‌افزار و پاک است پرداخته شد. لازم به‌ذکر است با توجه به تلاش‌های صورت گرفته به‌دلیل عدم دسترسی مجموعه داده استاندارد در این حوزه مجبور به تولید مجموعه‌داده مستقل شدیم. این طرح از سه مرحله تشکیل شده است. در مرحله اول مجموعه داده‌ی شامل فایل‌های اجرایی پاک و باج‌افزارها، در محیطی محافظت شده و با شرایط یکسان اجرا شده و تمام رفتارهای زمان اجرای آن‌ها نظارت شد. سپس به‌وسیله یک لاگر که توسعه داده شده و در برابر روش‌های دور خوردن آن توسط بدافزارها، مصون‌سازی شده است نسبت به استخراج گزارش فعالیت رفتاری نمونه‌ها اقدام شده است. ابزار مذکور که همان جعبه‌شن شخصی‌سازی شده Cuckoo است بر روی بستر مجازی‌سازی نصب و راه‌اندازی شده است. نتایج حاصل از اجرای نمونه فایل‌ها در قالب فایل Json ذخیره می‌شود. فراخوانی مرتبط با توابع فراخوانی سیستمی (API)، مقادیر بازگشتی، آرگومان‌های ورودی/خروجی، مقادیر رجیستری، رشته‌های متنی، ارتباطات شبکه‌ای و دیگر موارد مرتبط توسط این ابزار نظارت و ثبت می‌شود. پیش‌پردازشی روی خصیصه‌ها انجام می‌شود و موارد غیرمرتبط و اضافی غیر از توابع فراخوانی شده پاکسازی می‌شود.

در مرحله دوم و مهم‌ترین مرحله‌این طرح اقدام به نوشتن پارسر و سپس انتخاب ویژگی کردیم. چون هدف طرح مد نظر

شناسایی از طریق توابع سیستمی فراخوانی شده است لذا اقدام به نوشتن پارسری می‌کنیم که با تفسیر کردن فایل‌های Json حاوی رفتارهای حین اجرای مجموعه فایل‌های نمونه، به استخراج توابع فراخوانی سیستمی بپردازد و در انتها همه این توابع را در قالب یک فایل CSV در اختیار ما قرار دهد. قبل از آغاز مرحله انتخاب ویژگی یک مرحله پیش‌پردازش مورد نیاز است تا در آن بین تمامی توابع فراخوانی شده، اقدام به شناسایی توابعی کرده‌ایم که در اکثر باج‌افزارها مورد استفاده قرار گرفته است. این توابع برای تشخیص فایل‌های باج‌افزارها از بالاترین اولویت برخوردارند زیرا مختص باج‌افزارها هستند. این فهرست توابع در مراحل بعد برای انتخاب ویژگی‌های شناسایی کاربرد تأثیرگذاری خواهد داشت. در گام بعدی از این مرحله نسبت به انتخاب توابع مناسب به‌عنوان انتخاب ویژگی‌ها اقدام می‌کنیم. در این بخش روش‌های انتخاب ویژگی مورد مطالعه و بررسی قرار گرفته تا برای انتخاب بهترین توابع سیستمی جهت شناسایی دقیق‌تر باج‌افزارها مورد استفاده قرار گیرد. روش‌های انتخاب خصیصه جهت انتخاب زیرمجموعه‌ای کوچک‌تر از مجموعه اصلی خصیصه‌ها مد نظر قرار می‌گیرد.

در مرحله سوم و در آخرین مرحله‌این طرح نسبت به اخذ مدلی برای دسته‌بندی فایل‌های خوش‌خیم و باج‌افزارها از طریق یادگیری ماشین اقدام شد تا با اخذ بهترین مدل طبقه‌بندی روی ویژگی‌های اخذشده در مرحله قبل، نسبت به شناسایی باج‌افزارها عمل شود. در این مرحله نسبت به مطالعه و بررسی الگوریتم‌های دسته‌بندی یادگیری ماشین اقدام گردید.

همان‌طور که پیش‌تر بیان شد، نرخ تولید باج‌افزارها در سال‌ها و حتی ماه‌های گذشته به‌طور فزاینده‌ای رو به رشد هست. به‌نوعی می‌توان گفت حمله‌کنندگان سایبری میل و رغبت مضاعفی برای استفاده از باج‌افزارها در حملات خود داشته‌اند. در اثر این حملات هزینه‌های گزافی به قربانیان تحمیل شده است. هزینه‌هایی چون مالباختگی و مهم‌تر از همه سرقت، افشاء و یا پاک شدن اطلاعات مهم قربانیان ازجمله این خسارات است. لذا هدف اصلی این پژوهش بهینه‌سازی روش‌های شناسایی پیشین ارائه شده هست. این بهینه‌سازی مشتمل بر حالت بهینه در نرخ دقت تشخیص و سرعت آن است. چرا که هر چه سرعت و دقت تشخیص بالاتر باشد، در حجم‌های بالای تعداد باج افزارها نرخ شناسایی به‌مراتب افزایش می‌یابد.

۳-۱- تولید مجموعه داده

در گام ابتدایی این طرح نیاز است تا مجموعه‌ای از نمونه فایل‌های اجرایی شامل دسته‌ای از انواع باج‌افزارها و دسته‌ای از انواع فایل پاک تهیه کرد. لذا فرایند تولید مجموعه‌داده را به تفصیل در ادامه شرح داده‌ایم.

۳-۱-۱- تولید نمونه فایل‌های اجرایی

برای تهیه مجموعه نمونه فایل‌های باج افزار سعی کردیم از سطح اینترنت و سایت‌هایی چون ویروس ساین[1] یا مَل شیر[2] و یا ویروس‌توتال[3] و دیگر منابع مرتبط تهیه کنیم. در ادامه و با تحقیقات اقدام به نصب و راه‌اندازی ظرف عسل شدیم تا از طریق آن بتوانیم حملات شامل فایل‌های باج‌افزار را به دام اندازیم. در نهایت توانستیم به تعداد ۳۱۱ عدد نمونه فایل باج‌افزار از انواع خانواده باج افزارها تهیه نماییم. به تعداد ۶۰ عدد نمونه نیز برای مرحله ارزیابی در نظر گرفته شده است. دسته دوم از نمونه فایل‌های مجموعه‌داده مربوطه به فایل‌های اجرایی پاک و بی‌خطر تعلق دارد. این دسته فایل نیز که به تعداد ۳۹۰ عدد می‌باشد، نیز از سطح اینترنت جمع‌آوری شده است و شامل انواع دسته‌بندی‌های فایل اجرایی می‌باشد. تعداد ۶۷ نمونه فایل پاک نیز برای مرحله ارزیابی در نظر گرفته شده است. پس از فراهم‌سازی این مجموعه فایل‌ها، حال نیاز است که رفتارهای آن‌ها پس از اجرا تحت نظارت و عملیات مانیتورینگ قرار بگیرد.

۳-۱-۲- نظارت بر رفتار اجرای فایل‌ها

پس از تهیه مجموعه نمونه فایل‌های اجرایی شامل انواع خانواده باج افزارها و انواع دسته‌بندی فایل‌های اجرایی بی‌خطر، حال نسبت به اجرا و نظارت بر رفتارهای آن‌ها اقدام می‌کنیم. در ادامه ابتدا نحوه نظارت بر رفتار این فایل‌ها پس از اجرا که شامل تنظیمات محیط، اقدامات امن سازی محیط، نحوه اجرای فایل‌ها و استخراج گزارش‌ها است را تشریح می‌کنیم.

۳-۲- محیط اجرای فایل‌ها

همان‌طور که در پیش‌تر اشاره شد رویکرد این پژوهش مبتنی بر رفتار است. تحلیل مبتنی بر رفتار یا به‌نوعی تحلیل پویا چیزی جز اجرای برنامه‌ها نیست. لذا با توجه به ویژگی مخرب بودن و خرابه‌کارانه باج‌افزارها نباید آن‌ها را در محیط عادی و بدون حفاظت اجرا کرد؛ زیرا انجام این امر موجب تخریب نرم/سخت‌افزاری سیستم مربوطه، سرقت و از بین رفتن داده‌ها و دیگر موارد احتمالی گردد؛ بنابراین برای جلوگیری از تحقق خسارات احتمالی نیاز به محیط محافظت‌شده برای اجرای نمونه فایل‌های باج‌افزار با شرایط یکسان است. برای این منظور از بستر مجازی‌سازی Esxi نسخه ۵,۵ روی سرور استفاده نمودیم. سپس سامانه ثبت رخداد را روی یک میزبان با سیستم عامل Ubuntu 12.04 نصب کرده و برای اجرای فایل‌ها ۴ ماشین مجازی

[1] Virus Sign
[2] Malshare
[3] Virus Total

با سیستم عامل‌های ویندوز xp و ویندوز ۷ در نسخه‌های ۳۲ و ۶۴ بیتی راه‌اندازی نمودیم تا این مجموعه ابزارها برای اجرای نمونه فایل‌ها مهیا گردد و فرایند ثبت ویژگی‌ها آغاز گردد.

۳-۳- ابزارهای ثبت رفتار

پس از مهیاسازی محیط محافظت‌شده برای اجرای نمونه فایل‌ها حال باید ابزارهای مورد نیاز برای اجرای عملیات اخذ ویژگی‌های رفتاری حین اجرای فایل‌ها را نصب و راه‌اندازی نماییم. برای این منظور اقدام به نصب و راه‌اندازی جعبه شن کوکو نسخه ۲,۰ کردیم. این جعبه شن اقدام به اجرای فایل‌های نمونه پاک و باج‌افزار کرده و تمامی ویژگی‌ها را از زمان اجرای آن‌ها تا پس از اجرا را ثبت می‌کند. سپس تمام ویژگی‌های اخذ شده را در قالب فایل Json ارائه می‌دهد. خروجی‌های این جعبه‌شن بسته به نوع مأموریت و فعالیت فایل اجرایی دارای حجم فایل و تعداد ویژگی‌های متفاوت می‌باشد.

۳-۴- مقاوم‌سازی جعبه شن

یک نکته مهم و حائز اهمیت در این مرحله مقاوم‌سازی جعبه شن در برابر رفتارهای بدخواهانه باج‌افزارها است. برخی باج‌افزارها اقدام به تشخیص محیط‌های مجازی کرده و با توجه به شواهدی چون سیستم‌فایل‌های بسترهای مجازی‌سازی در مسیر رجیستری یا نصب و راه‌اندازی برخی خواص و ویژگی‌های این سیستم‌ها توان شناسایی محیط‌های مجازی را دارند. لذا نیاز است تا نسبت به جلوگیری از شناسایی توسط باج‌افزارها اقدامات مقاوم‌سازی را مد نظر قرار دهیم. به‌عنوان مثال چون ماهیت فعالیت باج‌افزارها بر مبنای عملیات خواندن و نوشتن است لذا این‌گونه بدافزارها در زمان اجرا روی فایل-های سیستمی جعبه شن اقدام به عملیات نوشتن و خواندن کرده و در نهایت منجر به عدم صحت در نتیجه ثبت رخدادها و گزارش نهایی می‌شد. لذا اقدام به پیشگیری و اعمال روش‌های عدم شناسایی محیط اجرا توسط باج‌افزارها گشته و در برخی موارد مجبور شدیم جهت اطمینان از صحت مرحله اخذ ویژگی‌ها از این‌گونه باج‌افزارها در مجموعه داده خود استفاده نکنیم.

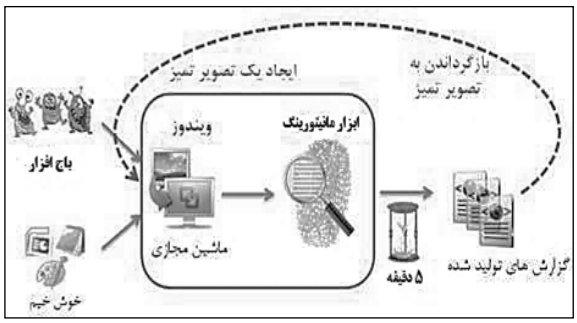
۳-۵- معیار مدت زمان محدود برای اجرای فایل

برخی باج‌افزارها رفتار خود را در زمان‌بندی خاصی بروز می‌دهند و در کد برنامه‌ای آن‌ها بر حسب نوع مأموریت و محیط مد نظر برای اجرای خود تایمرهای زمان‌بندی به‌کار گرفته شده است. در برخی موارد ممکن است یا باج‌افزار رفتار خود را پس از ۲۴ ساعت، چند روز، چند هفته و یا حتی چندین ماه و بیشتر بروز دهد. برای این‌گونه موارد زمان‌بندی ۵ دقیقه‌ای از زمان اجرا ملاک قرار داده شده است. بدین‌صورت که فایل‌هایی که پس از ۵ دقیقه اجرا و سپس منتهی به ایجاد گزارش نهایی نشوند حذف می‌شوند. معیار ۵ دقیق با توجه به زیرساخت مورد استفاده و آزمایش‌های میدانی روی غالب مجموعه داده، مد نظر قرار گرفته است. این معیار برای صحت هر چه بیشتر فایل گزارش نهایی مد نظر قرار گرفته است تا فایل گزارش رفتاری حاصل از اجرای باج‌افزارهایی با این ویژگی زمان‌بندی اجرا در فایل تجمیع شده نهایی ویژگی‌های رفتاری قرار نگرفته و ویژگی‌های اخذشده دارای صحت و دقت بیشتری باشد.

۳-۶- نحوه و شرایط انجام عملیات اخذ ویژگی

نکته قابل اشاره در این مرحله یکسان بودن شرایط برای اجرای هر یک از فایل‌های نمونه است. روند اجرای فایل‌ها بدین‌صورت است که ابتدا فایل اجرایی به‌عنوان ورودی به جعبه شن داده می‌شد و جعبه شن با توجه به درخواست و یا نیاز فایل ورودی محیط و سیستم‌عامل مربوطه را فراخوانی و اجرا می‌کرد. سپس قبل از اجرای فایل جعبه شن یک فایل پشتیبان جدید برای این فایل ورودی ایجاد می‌کرد. فایل ورودی را در سیستم‌عامل مد نظر اجرا کرده، نسبت به سیر مراحل نظارت اقدام می‌کرد.

در انتها و پس از اخذ گزارش رفتاری نهایی حاصل از اجرای فایل جعبه شن دوباره شرایط را از روی فایل پشتیبان به حالت اول بر می‌گرداند تا شرایط برای اجرای فایل بعدی مهیا و یکسان باشد. در برخی موارد فایل ورودی برای اجرا نیاز به یکسری فایل‌های سیستمی وابسته داشت که به‌صورت دستی این فایل‌ها را در مسیر مد نظر و مورد نیاز برنامه و فایل ورودی قرار دادیم تا سیر اجرای فایل ورودی خللی وارد نگردد. در شکل (۴-۲) روند مراحل شرح داده‌شده در فوق در قالب یک فرایند آورده شده است.



شکل (۱): روند نظارت و ثبت رخداد هنگام اجرای فایل اجرایی

در نهایت گزارش حاصل از اجرای فایل ورودی و تغییرات و رخدادهای زمان اجرای فایل در قالب فایل Json در اختیار کاربر قرار می‌گیرد.

۳-۷- ابزار پارسر

پس از اجرای تمامی نمونه فایل‌ها و اخذ تمامی گزارش‌های رفتاری حاصل از آن‌ها حال با توجه به اینکه موضوع این پژوهش

شناسایی بر مبنای توابع سیستمی فراخوانی شده است لذا باید از میان رخدادهای ثبت شده و اخذشده از اجرای نمونه فایل‌ها، توابع سیستمی را که در زمان اجرای فایل اجرایی فراخوانی شده‌اند را جداسازی کنیم و همه این توابع را در یک فایل تجمیع کنیم. از طرف دیگر به‌دلیل اینکه نیاز داریم روی فایل به‌دست آمده تجزیه و تحلیل آماری و داده‌کاوی انجام دهیم لذا نیاز است فایل نهایی در قالب یک فایلcsv باشد تا با برنامه‌های این زمینه سازگاری داشته باشد.

بر همین اساس نیاز داریم تا یک ابزار پارسر بنویسیم تا همزمان هم نسبت به جداسازی توابع فراخوانی شده سیستمی از میان همه ویژگی‌ها که در قالب فایل‌های Json می‌باشد اقدام شود هم در نهایت همه این توابع را در قالب یک فایل CSV با تفکیک تعداد فراخوانی شدن هر تابع اراده دهد.

۳-۸- انتخاب ویژگی

عملیات انتخاب ویژگی یک پیش‌پردازش برای فرایند داده‌کاوی محسوب می‌شود. با استفاده از مهارت‌های انتخاب ویژگی می‌توان به کاهش نرخ خطای مدل مورد آموزش به‌وسیله غربال ویژگی‌های اضافی و نامرتبط پرداخت. باید به این نکته توجه داشت که اگر چه در مرحله تولید ویژگی‌های حاصل از اجرای نمونه فایل‌های اجرایی، حجم بسیار عظیمی از ویژگی‌ها تولید می‌شود اما پس از بررسی‌ها می‌توان دریافت که حجم زیادی از این ویژگی‌ها در تعداد کمتری از فایل‌ها نسبت به کل نمونه فایل‌های اجرا شده وجود دارد. می‌توان گفت تعداد ویژگی‌های موجود پس از اجرای همه فایل‌های اجرایی مجموعه داده و موجود در کل گزارش‌های رفتاری اخذشده از آن‌ها حدود ۳۵۰۰۰۰ ویژگی بوده است. لذا با استفاده از مهارت‌های انتخاب خصیصه این ویژگی‌ها را غربال نماییم.

با مطالعات و بررسی‌های انجام‌شده بر روی طرح و بیشینه آن در قالب کارهای مرتبط، در دو قسمت می‌توان نسبت به شرایط بهینه اقدام نمود. یکی تهیه نمونه فایل‌ها برای اجرا و اخذ گزارش‌های رفتاری آن در قالب مجموعه داده و دیگری فایل یکپارچه حاصل از ثبت و ضبط توابع فراخوانی شده حین اجرای نمونه فایل‌ها است.

۳-۸-۱- انجام مرحله پیش‌پردازش قبل از انتخاب ویژگی

پس از اینکه توابع یا ویژگی‌های مورد استفاده از میان تمام ویژگی‌های رفتاری نمونه فایل‌های اجراشده در جعبه‌شن توسط پارسر استخراج و در یک فایل متمرکز گردید، در این بخش از میان توابع مستخرج اقدام به انتخاب توابعی کرده‌ایم که بیشترین شباهت را با توابع مورد استفاده باج‌افزارها دارند. در جدول زیر به این توابع اشاره شده است.

جدول (۱): توابع استخراج‌شده در مرحله پیش‌پردازش

ntclose	ntwritefile
ntunmapviewofsection	drawtextexw
ntreadfile	getfileattributesw
ntmapviewofsection	regqueryvalueexw
getsystemmetrics	loadstringw
readprocessmemory	ntdelayexecution

توابعی که در جدول (۱) مشاهده می‌شود، بیشترین فراخوانی را در میان دیگر توابع توسط نمونه فایل‌های باج‌افزار دارند. به عبارت دیگر توابع به‌دست‌آمده الگو و شاخص‌هایی هستند که تأثیر بسزایی در روند شناسایی باج‌افزارها در کنار شاخص‌ها و ویژگی‌های دیگر خواهند داشت.

۳-۸-۲- آزمایش‌ها چندمرحله‌ای آستانه تحمل عدم کاهش نرخ دقت تشخیص

در این قسمت عملیات غربال‌گری و حذف ویژگی‌ها را تا جایی ادامه می‌دهیم که از میزان نرخ دقت شناسایی کاسته نشود. به‌نوعی می‌توان گفت آستانه تحمل مجموعه داده در عدم کاهش نرخ دقت تشخیص را می‌سنجیم. در این روند ویژگی‌هایی که در بخش قبلی بر اساس فراوانی فراخوانی توابع مرتب‌سازی شده، از بیشترین میزان تأثیرگذاری تا کمترین میزان مرتب شده است را در نظر گرفته و از انتهای این ویژگی‌ها یعنی ویژگی‌هایی که بار کمتری روی نرخ دقت تشخیص دارند، ویژگی‌ها را با نرخ معینی کم کرده و هر بار نرخ دقت تشخیص را به‌وسیله الگوریتم‌های دسته‌بندی مد نظر محاسبه کرده تا نرخ دقت تشخیص کنترل شود. در نهایت به آستانه یا نقطه عطفی می‌رسیم که اگر از آنجا به بعد ویژگی را از مجموع ویژگی‌ها بکاهیم، نرخ دقت شروع به کاهش خواهد کرد. این روش نیز برای اطمینان از بهینه‌سازی کامل مجموعه داده ما نسبت دیگر مجموعه داده‌ها است.

۳-۹- آموزش ویژگی‌ها با استفاده از الگوریتم‌های دسته‌بندی

پس از اینکه بهترین ویژگی‌ها برگزیده شدند، در این بخش الگوریتم‌های داده‌کاوی مورد استفاده برای دسته‌بندی فایل‌های باج‌افزار از فایل‌های پاک شرح داده می‌شوند. با استفاده از این الگوریتم‌ها یافتن الگوهای تفکیک کننده در مجموعه داده و یا ویژگی‌های آن میسر می‌شود. پس از کشف الگوهای مناسب و

متناسب با مجموعه داده، این الگوها برای دسته‌بندی نمونه فایل‌های ناشناخته مورد استفاده قرار می‌گیرد. با توجه به اینکه تنها دو دسته یا دو کلاس فایل باج‌افزار و پاک داریم لذا از الگوریتم‌های دسته‌بندی دودویی برای شناسایی آن‌ها استفاده می‌کنیم.

هدف اصلی، دسته‌بندی فایل‌های پاک از باج‌افزارها با نرخ دقت بالا است. به همین منظور ورودی این الگوریتم‌ها فایل ویژگی‌های بهینه‌شده در مرحله قبل می‌باشد. الگوریتم‌های درخت تصمیم J48، جنگل تصادفی، NaiveBayes، لاجستیک ساده، SMO، کرنل خطی، پایه شعاعی در این پژوهش مورد استفاده قرار گرفته است. این الگوریتم‌ها به‌وسیله فرآیند اعتبارسنجی ۱۰ قسمتی برای جلوگیری از احتمال هم‌پوشانی بیش از حد مدل مورد ارزیابی استفاده شده است.

۴- ارزیابی و تحلیل نتایج

در این بخش به بررسی و صحت‌سنجی روش پیشنهادی ارائه شده در این پایان‌نامه خواهیم پرداخت. لذا در ادامه به بررسی نتایج حاصل از آزمایش ورودی‌های این طرح در قالب جداول و تصاویر آن می‌پردازیم. جزئیات داده‌های مورد استفاده، معیارهای ارزیابی، سؤالات مطرح شده و پاسخ به آن‌ها در این فصل شرح داده خواهد شد.

۴-۱- مجموعه داده‌ها و تنظیمات آزمایش‌ها

در این پژوهش دو مجموعه‌داده مورد استفاده قرار گرفته است. مجموعه‌داده ابتدایی که شامل فایل‌های اجرایی پاک و باج‌افزار است. این فایل‌ها از فضای اینترنت وب سایت‌هایی چون ویروس‌ساین یا ملشیر و یا ویروس‌توتال جمع‌آوری شده است. در تهیه بخش فایل‌های باج‌افزار تلاش بر این بوده است که برای صحت بیشتر نتایج طرح و فراگیر بودن و جامع بودن آن، از گونه‌های مختلف باج‌افزارها شامل خانواده متعدد باج‌افزارها مانند Cerber، Cryptolocker، Cryptowall، Locky، Sage، wannacry، Torrentlocker استفاده شده است. به‌علت کمبود نمونه فایل‌های باج‌افزار در فضای اینترنت و نیاز به تأمین این فایل‌ها و نیز عدم دسترسی به خرید و دانلود بانک بدافزارها از شرکت‌های امنیتی سایبری به‌علت تحریم‌ها، بنا به تجربه قبلی اقدام به نصب و راه‌اندازی ظرف‌عسل یا تله سایبری [1] در مسیرهای اینترنتی آلوده شده و توسط آن برخی از فایل‌های مجموعه‌داده را تهیه کردیم. در نهایت توانستیم به تعداد ۳۱۱ عدد نمونه فایل باج‌افزار از انواع خانواده آن تهیه کنیم.

بخش دوم مجموعه‌داده شامل فایل‌های پاک قابل حمل است. این بخش از نمونه فایل‌ها نیز از دسته‌بندی‌های مختلف برنامه‌ها تهیه شده است. از جمله این دسته‌بندی‌ها می‌توان به چندرسانه‌ای، اسنادی، امنیتی، گرافیکی، نظارتی، بازی و فایل‌های سیستمی اشاره کرد. این فایل‌ها اغلب به‌دلیل قابل حمل بودن دارای حجم پایین می‌باشد. در نهایت تعداد ۳۹۰ فایل پاک برای مجموعه‌داده ابتدایی این طرح جمع‌آوری شده است.

از حجم کل این مجموعه داده حدود ۳۰ درصد آن به مجموعه نهایی برای استفاده در مرحله ارزیابی تعلق گرفته است. در جدول (۲) جزئیات مجموعه‌داده ابتدایی و نهایی به همراه داده‌های آماری آن‌ها نشان داده شده است.

جدول (۲): میزان فراوانی نمونه فایل‌های باج افزار نسبت به خانواده باج‌افزارها

خانواده‌های باج‌افزار	ارزیابی	آموزش
Cerber	۱۰	۴۰
Cryptolocker	۸	۳۴
Cryptowall	۵	۳۱
Locky	۱۳	۴۵
Sage	۵	۲۷
wannacry	۱۴	۵۳
Torrentlocker	۵	۲۱

جدول (۳) میزان فراوانی فایل‌های بی‌خطر جمع‌آوری‌شده نسبت به دسته‌بندی‌های متعدد آن را نشان می‌دهد.

جدول (۳): فراوانی نمونه فایل‌های بی‌خطر نسبت به حوزه‌های مختلف

دسته‌بندی فایل‌های پاک	ارزیابی	آموزش
چندرسانه‌ای	۸	۳۹
اسنادی	۱۰	۴۹
امنیتی	۸	۴۶
گرافیکی	۷	۳۹
نظارتی	۱۳	۵۷
بازی	۵	۲۹
سیستمی	۱۶	۶۴

در جدول (۳) می‌توان میزان فراوانی نمونه فایل‌های پاک قابل‌حمل و باج‌افزار را نسبت به کل مجموعه‌داده در مراحل ارزیابی و آموزش مشاهده نمود.

[1] Honeypot

۴-۲- مرحله انتخاب ویژگی

در این مرحله که مهم‌ترین بخش این طرح می‌باشد، باید عملیات انتخاب ویژگی صورت گیرد. این انتخاب باید به‌گونه‌ای باشد تا ویژگی‌های منتخب منجر به ایجاد بهینه‌سازی در سرعت و دقت نرخ تشخیص باج‌افزارها شود. لذا به همین منظور و برای ارزیابی و مقایسه مقادیر این پژوهش با طرح هری کریشنان و سومن [۱۰] مقایسه می‌شود. دلیل این مقایسه، قرابت مجموعه‌داده این طرح از نظر محتوا و زمان اجرای آن با این پژوهش است. چون در میان پژوهش‌های قبلی طرح مذکور بیشترین شباهت با طرح این مقاله از منظر توابع مورد استفاده، سیستم‌عامل مد نظر، تا حدی ابزار به‌کاربرده شده و دیگر موارد را داشت لذا یکی از معیارهای مقایسه را برای اثبات بهبود یافتن طرح این مقاله، طرح هری کریشنان و سومن قرار دادیم. در مقاله طرح کریشنان ویژگی‌های نهایی طرح مندرج شده و قابل مقایسه با ویژگی‌های این طرح نیز بود.

چون مبنای ارزیابی این طرح ویژگی‌های طرح کریشنان و سومن است و در این طرح تعداد ویژگی‌های مجموعه‌داده آن ۱۲۵ عدد است لذا ویژگی‌های فعلی اخذ شده مورد استفاده در این پایان‌نامه ۲۳۲ عدد است و باید نسبت به بهینه‌سازی تشخیص آن‌که شامل نرخ سرعت و دقت تشخیص است، اقدام کنیم.

ادامه برای انتخاب مؤثرترین ویژگی‌ها از چند الگوریتم انتخاب ویژگی مرتبط با مجموعه داده این پژوهش استفاده کرده و نتایج حاصله را ارائه می‌دهیم.

۴-۲-۱- استفاده از الگوریتم انتخاب ویژگی Gain Ratio

در ابتدای گام بهینه‌سازی از الگوریتم gain ratio برای انتخاب ویژگی مورد استفاده قرار دادیم. این الگوریتم نسبت اطلاعات به‌دست آمده از ویژگی‌ها به اطلاعات ذاتی خود ویژگی را محاسبه می‌کند. پس از اعمال الگوریتم انتخاب ویژگی GainRatio روی مجموعه‌داده تعداد ۱۱۷ ویژگی از میان ۲۳۲ ویژگی بر اساس این الگوریتم انتخاب گردید. البته باید اشاره کرد بر اساس معیارهای این الگوریتم به هر ویژگی این مجموعه داده میزان یا نرخی اختصاص می‌گیرد که خروجی محاسباتی آن شامل یک فهرست از ویژگی‌ها است که بر اساس این نرخ‌ها مرتب‌سازی شده است و طبق آن ویژگی‌هایی که نرخ آن‌ها کمترین میزان را دارند (یعنی کمترین تأثیرگذاری را دارند) از این فهرست حذف می‌شوند. حال باید میزان نرخ دقت تشخیص این ویژگی‌های به‌دست‌آمده را به‌وسیله الگوریتم‌های یادگیری ماشین مد نظر قرار داده و محاسبه کنیم. لازم به‌ذکر است معیار این بررسی و نتایج حاصل از آن بر مبنای نرخ دقت تشخیص، نرخ مثبت کاذب، نرخ مثبت صحیح و مدت زمان تشخیص می‌باشد. به جدول (۴) دقت کنید.

جدول (۴): نرخ دقت تشخیص به‌دست‌آمده مهارت gain ratio به‌وسیله الگوریتم‌های دسته‌بندی مد نظر

الگوریتم دسته‌بندی	نرخ مثبت کاذب	نرخ مثبت صحیح	نرخ دقت تشخیص	زمان (ثانیه)
بیز ساده	۰/۱۶۹	۰/۷۹۴	۷۹٪/۴۰	۰/۱۳
لاجستیک ساده	۰/۱۲۳	۰/۸۶۶	۸۶٪/۵۶	۰/۸۹
J48	۰/۱۰۷	۰/۸۸۸	۸۸٪/۸۳	۰/۱۹
جنگل تصادفی	۰/۰۵۶	۰/۹۴۲	۹۴٪/۲۴	۱/۵۲
SMO	۰/۱۸۴	۰/۷۸۲	۷۸٪/۱۸	۰/۱۴
کرنل خطی	۰/۱۷۹	۰/۸۳۱	۸۳٪/۰۷	۰/۳۱
پایه شعاعی	۰/۱۷۴	۰/۸۵۹	۸۲٪/۸۳	۰/۲۲

با توجه به نتایج حاصل‌شده در جدول (۴) مشاهده می‌شود که بیشترین میزان دقت تشخیص در میان نتایج بالا مربوط به الگوریتم جنگل تصادفی با نرخ دقت تشخیص ۹۴/۲۴٪ و نرخ مثبت صحیح ۰/۹۴۲ و نرخ مثبت کاذب ۰/۰۵۶ با مدت زمان تشخیص ۱/۵۲ ثانیه می‌باشد. همان‌طور که مشاهده می‌شود این الگوریتم از نرخ دقت تشخیص بالایی برخوردار است اما به مراتب زمان تشخیص بالایی نیز دارد.

۴-۲-۲- استفاده از الگوریتم انتخاب ویژگی همبستگی

پس از اعمال الگوریتم انتخاب ویژگی همبستگی روی مجموعه‌داده تعداد ۱۷۰ ویژگی از میان ۲۳۲ ویژگی بر اساس این الگوریتم انتخاب گردید. البته باید اشاره کرد بر اساس معیارهای این الگوریتم به هر ویژگی این مجموعه داده میزان یا نرخی اختصاص می‌گیرد که خروجی محاسباتی آن شامل یک فهرست از ویژگی‌ها است که بر اساس این نرخ‌ها مرتب‌سازی شده است و طبق آن ویژگی‌هایی که نرخ آن‌ها کمترین میزان را دارند (یعنی کمترین تأثیرگذاری را دارند) از این فهرست حذف می‌شوند. حال باید همانند الگوریتم قبلی میزان نرخ دقت تشخیص این ویژگی‌های به‌دست آمده را به‌وسیله الگوریتم‌های یادگیری ماشین مد نظر قرار داده و محاسبه کنیم. لازم به‌ذکر است معیارهای حساسیت در این بررسی و نتایج حاصل از آن بر مبنای نرخ دقت تشخیص، نرخ مثبت کاذب، نرخ مثبت صحیح و مدت زمان تشخیص مد نظر قرار گرفته است. جدول (۵) نرخ‌های معیارهای مذکور را بر اساس الگوریتم‌های دسته‌بندی مد نظر در این طرح به نمایش کشیده است.

جدول (۵): نتایج حاصل‌شده از اعمال الگوریتم انتخاب ویژگی همبستگی

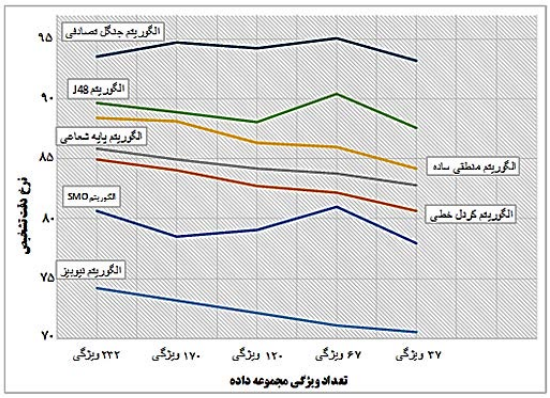
الگوریتم دسته‌بندی	نرخ مثبت کاذب	نرخ مثبت صحیح	نرخ دقت تشخیص	زمان (ثانیه)
بیز ساده	۰/۴۵۷	۰/۶۳۲	٪۶۳/۱۷	۰/۰۸
لاجستیک ساده	۰/۱۳۴	۰/۸۵۳	٪۸۵/۳۴	۱/۱۴
J48	۰/۱۱۲	۰/۸۸۳	٪۸۸/۳۰	۰/۲۵
جنگل تصادفی	۰/۰۶۰	۰/۹۳۷	٪۹۳/۷۱	۱/۲
SMO	۰/۱۶۱	۰/۸۰۵	٪۸۰/۴۵	۰/۱۷
کرنل خطی	۰/۱۴۳	۰/۸۵۵	٪۸۵/۵۱	۱/۰۱
پایه شعاعی	۰/۱۷۴	۰/۸۵۹	٪۸۵/۸۶	۰/۲۸

اگر نتایج حاصل‌شده در جدول (۵) را مورد ارزیابی قرار دهیم، مشاهده می‌کنیم که نرخ‌های به‌دست آمده دارای ویژگی‌های خاص به خود هستند و این بر اساس نوع الگوریتم دسته‌بندی می‌باشد. برخی از الگوریتم‌های دسته‌بندی مانند بیز ساده بسیار سبک بوده و دارای سرعت بالایی است که در اینجا زمان تشخیص آن ۰/۰۸ ثانیه برآورد شده است اما دارای نرخ دقت تشخیص ضعیف ٪۶۳/۱۷ می‌باشد. برخی دیگر از الگوریتم‌ها مانند جنگل تصادفی دارای نرخ دقت تشخیص بالا هستند مثلاً در اینجا ٪۹۳/۷۱، اما ماهیت سنگینی برای پیاده‌سازی دارند زمان بالاتری می‌طلبند که در اینجا ۱/۲ ثانیه می‌باشد.

۴-۲-۳- روش انتخاب ویژگی با محاسبه فراخوانی توابع و استخراج نقطه بهینه نرخ دقت و سرعت در شناسایی

همان‌طور که در قسمت‌های قبلی شرح داده شد دو الگوریتم انتخاب ویژگی برای بهینه‌سازی بیشتر روی مجموعه‌داده اعمال شده و ویژگی‌های استخراج‌شده بر مبنای این دو الگوریتم نمایش داده شد. در ادامه اقدام به اعمال الگوریتم‌های دسته‌بندی روی ویژگی‌های به‌دست آمده کردیم و نرخ‌های معیارهای تشخیص را به‌دست آوردیم که در قالب جداول ارائه گردید.

در ادامه روند بهینه‌سازی مجموعه‌داده اقدام به انتخاب ویژگی‌ها در چند مرحله به‌وسیله محاسبه نرخ فراوانی فراخوانی‌های توابع یا ویژگی‌ها نمودیم. اینکه در چند مرحله اقدام به استخراج این ویژگی‌ها شده است به این دلیل است تا بتوانیم آستانه تحمل نرخ دقت و سرعت در شناسایی را محاسبه و استخراج نماییم. برای این مهم ابتدا فهرستی مرتب‌سازی شده از تعداد فراخوانی توابع این مجموعه‌داده تهیه می‌نماییم. سپس در هر مرحله از انتهای این فهرست و توابعی که تعداد فراخوانی آن‌ها کمتر است را حذف می‌نماییم. با این کار از تعداد ویژگی‌های کل مجموعه‌داده کاسه می‌شود و توقع می‌رود زمان تشخیص کاهش یافته باشد. از طرفی باید در هر مرحله نرخ دقت در تشخیص را ارزیابی نموده تا میزان این نرخ کاهش نیابد. با تکرار این کار به مرحله‌ای می‌رسیم که با کاهش تعداد ویژگی‌های مجموعه‌داده نرخ دقت تشخیص نیز کاهش می‌آید و آنجا همان آستانه تحمل نرخ دقت و سرعت تشخیص این مجموعه‌داده یا به عبارتی نقطه عطف و تلاقی این نرخ می‌باشد. در جداول زیر نتایج به‌دست‌آمده از ارزیابی مجموعه‌داده در هر مرحله به تفکیک الگوریتم‌های دسته‌بندی به تصویر کشیده شده است. در جدول (۴) نتایج ارزیابی بر اساس الگوریتم دسته‌بندی بیز ساده آمده است.



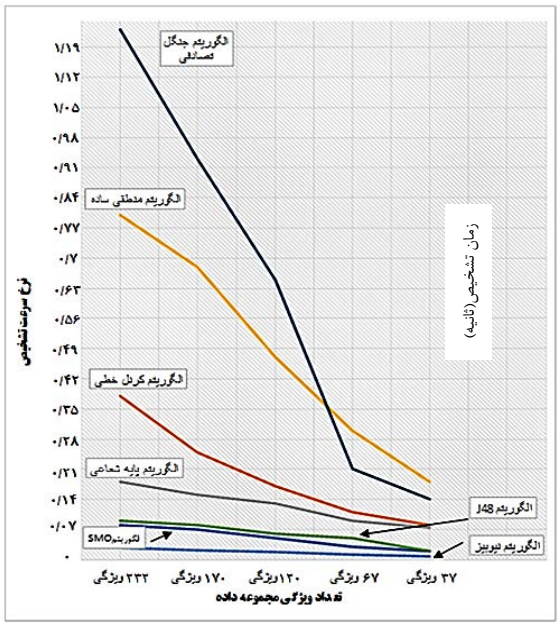
نمودار (۱): نتایج حاصل از ارزیابی ۵ مرحله‌ای تعداد ویژگی‌های مجموعه داده بر مبنای نرخ دقت

با توجه به نتایج حاصل از ارزیابی‌ها در نمودار (۱)، مشاهده می‌شود که در اکثر الگوریتم‌های دسته‌بندی نقطه نزولی نرخ دقت تشخیص، نقطه‌ای است که ویژگی‌های مجموعه‌داده از عدد ۶۷ روند کاهشی را آغاز می‌کند. این اثبات‌کننده این است که تعداد بهینه ویژگی برای مجموعه‌داده تعداد ۶۷ است. نکته دیگر حائز اهمیت این است که با توجه به نمودار مشاهده می‌شود که الگوریتم‌های دسته‌بندی درخت تصمیم یعنی جنگل تصادفی و J48 نسبت به دیگر الگوریتم‌ها از تناسب بهتری با مجموعه‌داده این طرح دارند و از نرخ دقت بالاتری نیز برخوردارند.

پس از بررسی و ارزیابی نتایج نرخ دقت تشخیص، به بررسی نرخ سرعت تشخیص بر مبنای الگوریتم‌های دسته‌بندی مذکور می‌پردازیم. در ابتدا به نمودار (۲) توجه کنید.

همان‌طور که در نمودار (۲) قابل مشاهده است، به‌طور طبیعی زمان تشخیص در اکثر الگوریتم‌ها با کاستن از ویژگی‌ها رو به کاهش می‌باشد. این نرخ نزولی سرعت تشخیص در الگوریتم‌های دسته‌بندی حوزه رگرسیون و نیز الگوریتم J48 با سرعت کمتری می‌باشد. دلیل آن هم این است که این دسته الگوریتم‌های دسته‌بندی دارای ساختاری سبک و چابک‌تر و در نتیجه سریع‌تر از الگوریتم‌های درخت تصمیم است و در نتیجه نرخ نزولی سرعت در این دسته از الگوریتم‌ها کمتر محسوس است؛ اما نرخ نزولی سرعت در الگوریتم جنگل تصادفی با شیب

تندی قابل ملاحظه است و این نشان می‌دهد با اعمال عملیات حذف ویژگی‌های چندمرحله‌ای روی مجموعه‌داده بر اساس الگوریتم جنگل تصادفی شاهد افزایش نرخ سرعت و در نتیجه تسریع در عملیات شناسایی خواهیم بود. پس اگر کمیت مجموعه‌داده ما چندین برابر مجموعه‌داده فعلی باشد با اعمال عملیات بهینه‌سازی ویژگی‌ها روی مجموعه‌داده بر اساس الگوریتم جنگل تصادفی نتیجه بهینه روی نرخ سرعت تشخیص حاصل می‌شود.



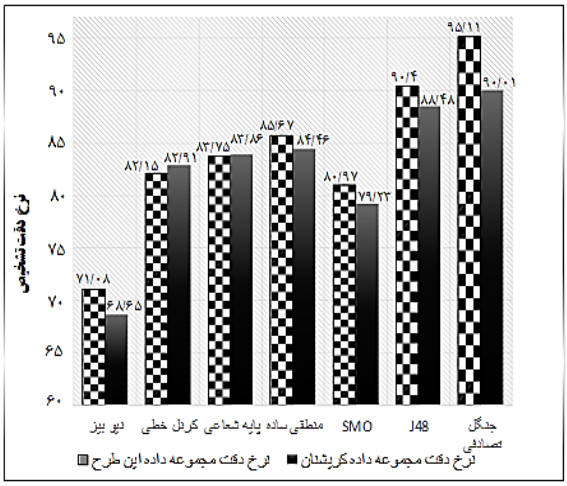
نمودار (۲): نتایج حاصل از ارزیابی ۵ مرحله‌ای تعداد ویژگی‌های مجموعه داده بر مبنای زمان تشخیص

با توجه به نتایج به‌دست‌آمده در پنج مرحله آزمایش روی مجموعه‌داده برای استخراج نرخ بهینه سرعت و دقت در تشخیص به این نتیجه می‌رسیم که تا رسیدن به مجموعه‌داده با تعداد ۶۷ ویژگی نرخ دقت تشخیص در بالاترین حد ممکن خود قرار دارد اما پس از انجام آخرین آزمایش و تقلیل تعداد ویژگی‌ها به تعداد ۳۷ عدد، شاهد کاهش چشمگیر نرخ دقت تشخیص بوده‌ایم. لذا می‌توان گفت که آستانه تحمل و نقطه عطف این مجموعه‌داده برای کشف نرخ بهینه دقت و سرعت در تشخیص، مجموعه داده با ۶۷ ویژگی است.

۴-۳- تحلیل و ارزیابی مرحله انتخاب الگوریتم دسته‌بندی

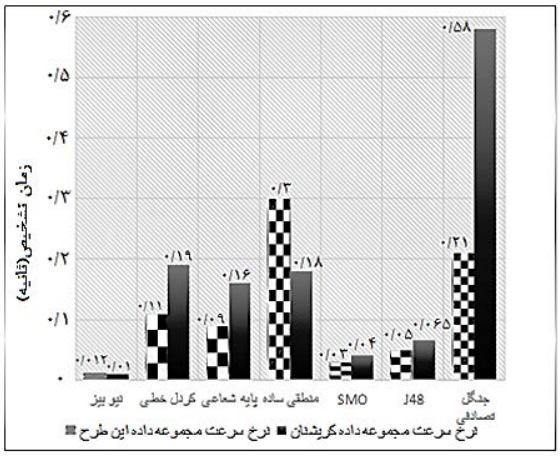
در بخش قبل با تحلیل و ارزیابی مراحل ۵ گانه آزمایشی روی مجموعه‌داده، به مجموعه‌داده بهینه شامل ۶۷ ویژگی دست پیدا کردیم. حال در این بخش به تحلیل و ارزیابی نتایج برای انتخاب بهترین مدل دسته‌بندی می‌پردازیم.

با توجه با نتایج حاصل‌شده از اعمال انواع الگوریتم‌های دسته‌بندی روی مجموعه‌داده می‌توان دریافت که الگوریتم‌های حوزه درخت تصمیم از آمار نرخ تشخیص بهتری نسبت به دیگر الگوریتم‌ها برخوردارند. با ارزیابی دقیق‌تر نسبت به نتایج حاصل شده و بر مبنای معیارهایی چون نرخ دقت تشخیص، نرخ مثبت کاذب، نرخ مثبت صحیح و در نهایت مدت زمان اجرا می‌توان گفت الگوریتم جنگل تصادفی با نرخ دقت تشخیص ۹۵/۱۱٪، نرخ مثبت صحیح ۰/۹۵۱، نرخ مثبت کاذب ۰/۰۴۷ و در نهایت با سرعت تشخیص ۰/۲۱ ثانیه برای این مجموعه‌داده با ۶۷ ویژگی الگوریتم بهینه دسته‌بندی است. لازم به‌ذکر است که طرح متناظر کریشنان با الگوریتم دسته‌بندی پایه‌شعاعی دارای نرخ دقت تشخیص ۸۳/۸٪، نرخ مثبت صحیح ۰/۸۷۷ و نرخ مثبت کاذب ۰/۰۷۵ و مدت زمان تشخیص ۰/۶۱ ثانیه (طبق پیاده‌سازی ویژگی‌های اعلام‌شده در طرح و محاسبه زمان تشخیص بر مبنای الگوریتم دسته‌بندی مربوطه) روی مجموعه‌داده با ۱۲۰ ویژگی بوده است. در بهترین حالت نرخ دقت تشخیص مجموعه‌داده طرح کریشنان و سومن، که در این پژوهش آن را بر مبنای ویژگی‌های مجموعه‌داده مندرج در آن طرح و نیز بر اساس الگوریتم جنگل تصادفی ارزیابی کردیم؛ نرخ دقت معادل ۹۰/۰۱٪، مثبت کاذب معادل ۰/۱۰۵، نرخ مثبت صحیح معادل ۰/۷۸۰ و سرعت تشخیص معادل ۰/۹۸ ثانیه بوده است، که نسبت به نتایج نرخ تشخیص این پژوهش به مراتب دارای معیارهایی با نرخ‌های کمتری بوده است. در جدول زیر نتایج ارزیابی بین این دو طرح برای مقایسه و درک بهتر آورده شده است. به نمودار مقایسه‌ای ۳ که گویای ادعای مذکور است دقت کنید.



نمودار (۳): نمودار مقایسه‌ای نرخ دقت تشخیص طرح کریشنان و طرح این پژوهش

در ادامه و در نمودار (۴) می‌توان نرخ سرعت تشخیص در هر دو طرح کریشنان و این پژوهش را نیز مشاهده نمود.



نمودار (۴): مقایسه‌ای زمان تشخیص طرح کریشنان و طرح این پژوهش

همان‌طور که در نمودار (۴) مشاهده می‌شود، زمان تشخیص در طرح کریشنان با شاخص‌های میله‌ای سیاه رنگ، در اکثر الگوریتم‌ها بالاتر از شاخص‌های میله‌ای شطرنجی طرح این مقاله می‌باشد. لذا سرعت تشخیص در طرح این مقاله بالاتر از طرح کریشنان است.

با توجه به نتایج مورد مشاهده در نمودارهای (۳ و ۴)، می‌توان ارزیابی و مقایسه بهتری در مورد نرخ‌های دقت و سرعت در تشخیص بین طرح این طرح و طرح کریشنان داشت. همان‌طور که مشاهده می‌شود، طرح پیشنهادی این پایان‌نامه با نرخ دقت ۹۵/۱۱ درصد در مدت ۰/۲۱ ثانیه نسبت به نرخ دقت ۹۰/۰۱ درصد در مدت ۰/۵۸ ثانیه طرح کریشنان و سومن دارای نرخ تشخیص بهینه می‌باشد.

۵- نتیجه‌گیری

در این پژوهش یک روش بهینه کاربردی برای شناسایی پویای باج‌افزارها ارائه شد. ویژگی‌های استفاده‌شده در این طرح توابع فراخوانی شده حین اجرای فایل‌ها بود. طبق آزمایش‌ها صورت گرفته مشخص شد که این ویژگی‌ها قابلیت تمایز بالایی نسبت به همدیگر دارند و این به‌دلیل استفاده از نمونه‌های متنوع فایل‌های مربوطه بوده است. در این طرح نشان دادیم که توابع فراخوانی شده استخراج‌شده از عملکرد رفتاری فایل‌ها به خوبی می‌توانند عملکرد و رفتار فایل‌ها را مدل کنند.

طی نتایج حاصل از آزمایش‌ها و بر طبق مطالعات صورت گرفته روی کارهای مرتبط این طرح به نظر می‌رسد بیشتر پژوهشگران این عرصه تمرکز خود را بر روی افزایش نرخ دقت تشخیص قرار داده‌اند و نکته حائز اهمیت کاهش نرخ سرعت تشخیص و در نتیجه کاهش منابع پیاده‌سازی طرح شناسایی مغفول واقع شده است. به همین دلیل در این طرح به این نکته مهم پرداخته و تمرکز خود را برای کاهش مدت زمان تشخیص در عین حفظ بالا بودن نرخ دقت تشخیص قرار دادیم. لذا با انجام آزمایش‌ها متعدد روی کاهش ویژگی‌های زائد و کم اثر مجموعه‌داده به این مهم پرداختیم. آزمایش‌ها نشان می‌دهد که صرف نظر از مجموعه‌داده، ویژگی‌های تولید شده قابلت تمایز بالایی دارند. این مهم به‌وسیله در نظر گرفتن دو مجموعه‌داده ابتدایی و نهایی و نیز نوع جدید از باج‌افزارها به آن افزوده شده بود، مورد بررسی قرار گرفت. طبق نتایج آزمایش‌ها با وجود مشاهده رفتارهای جدید در قالب افروده شدن فایل‌های باج‌افزار جدید در مجموعه‌داده نهایی، ویژگی‌های اخذ شده در مجموعه‌داده اولیه که مدل شده‌اند قادر به شناسایی باج‌افزارهای جدید هستند. همچنین طی نتایج آزمایش‌ها، با به‌روزرسانی ویژگی‌های جدید و مدل‌سازی آن‌ها دقت تشخیص افزایش میابد. به‌روزرسانی ویژگی‌ها به‌دلیل مرور زمان و مشاهده رفتارهای ناشناخته باج‌افزارها امری اجتناب‌ناپذیر و منطقی است.

در این طرح در چند مرحله و پس از مطالعه و ارزیابی ابتکارات و نوآوری‌های مورد نیاز به شرح زیر صورت گرفت.

- ایجاد یک مجموعه داده مستقل به‌وسیله انواع خانواده باج‌افزارها و انواع دسته‌های متعدد فایل‌های پاک برای جامع بودن هر چه بیشتر نتایج پژوهش.
- اعمال یک مرحله فیلتر حذف توابع سیستمی غیرمرتبط و همچنین در نظر گرفتن یک مجموعه توابع سیستمی مرتبط و مؤثر در نرخ تشخیص با توجه به مطالعات صورت گرفته بر روی رفتار و عملکرد باج‌افزارها و استفاده از مشورت متخصصین و تحلیلگران بدافزار.
- استخراج مجموعه داده بهینه از نظر نرخ دقت و سرعت تشخیص به‌وسیله اعمال ارزیابی ۴ مرحله‌ای محاسبه نرخ فراوانی فراخوانی توابع سیستمی و کاهش ۱۲۶ ویژگی به ۶۷ ویژگی بدون کاسته شدن از نرخ دقت که موجب سبک و چابک شدن مجموعه‌داده می‌گردد.

می‌توان از مهم‌ترین دستاوردهای این پژوهش به موارد زیر اشاره داشت:

- تولید مجموعه‌داده استاندارد و بهینه باج‌افزارها
- استفاده از ۶۷ ویژگی بجای ۱۲۶ ویژگی در مجموعه‌داده بدون تغییر در نرخ دقت برای شناسایی باج‌افزارها
- استخراج ویژگی‌های خاص منظوره و پرکاربرد باج افزارها
- احصاء روش بهینه‌سازی نرخ سرعت و دقت تشخیص باج‌افزارها

در ادامه این طرح می‌توان برای رشد و توسعه این عرصه روی محدودیت‌های این طرح متمرکز شد و در جهت رفع آن گام برداشت.

• نکته اول نوشتن یک برنامه برای ثبت بهتر رخدادهای رفتاری حین اجرای فایل‌ها با پارامترهای بیشتر و مؤثرتر است. به‌عنوان مثال نوشتن یک لاگر مقاوم‌سازی شده در برابر روش‌های هوشمندانه باج‌افزارها مثل شناسایی محیط مجازی یا محیط دیباگ و دیگر موارد است.

• نکته دیگر کاربر روی باج‌افزارهای سیستم‌عامل‌های دیگری چون لینوکس یا آندروید است که به جامعیت این طرح می‌افزاید.

• کار دیگری که می‌توان به‌عنوان پیشنهاد در این بخش مطرح کرد اعمال روش ان-گرام برای ارزیابی نتایج بهینه‌سازی نرخ تشخیص حاصل از آن روی مجموعه‌داده این طرح می‌باشد.

• از دیگر مواردی که می‌توان به‌عنوان پیشنهاد برای کار روی این پژوهش مطرح کرد، استفاده از قوانین انجمنی برای یافتن ارتباط احتمالی بین ویژگی‌های مجموعه‌داده در بخش انتخاب ویژگی و یا بخش غربالگری ویژگی‌های مجموعه‌داده است.

• در آخرین پیشنهاد برای کارهای آینده این طرح می‌توان به خودکارسازی تمام مراحل این طرح اشاره داشت تا بتوان این طرح را در قالب یک محصول ارائه داد.

۶- مراجع

[1] ACSC, Threat Report 2017, p. 40, Jan. 2017. [online], available: https://www.cyber.gov.au/sites/default/files/2019-03/ACSC_Threat_Report_ 2017.pdf

[2] IOCTA, "Internet Organised Crime Threat Assessment (IOCTA)," 2017. available: https://www.europol.europa.eu/sites/default/files/documents/iocta2017.pdf.

[3] Symantec Corporation, "2018 Internet security threat report," vol. 23, pp. 1–89, 2018.

[4] D. Palmer, " how bitcoin helped fuel an explosion in ransomware attacks, " www.zdnet.com, 22 August 2016. [Online]. Available: https://www.zdnet.com/article/how-bitcoin-helped-fuel-an-explosion-in-ransomware-attacks/. [Accessed 12 Nov 2016].

[5] M. Hopkins and A. Dehghantanha, "Exploit Kits: The production line of the Cybercrime economy?," 2015 2nd Int. Conf. Inf. Secur. Cyber Forensics, Info Sec 2015, pp. 23–27, 2016.

[6] L. Usman, Y. Prayudi, and I. Riadi, "Ransomware analysis based on the surface, runtime and static code method," Apple inc. Inf. Technol., vol. 95, no. 11, pp. 2426–2433, 2017.

[7] Y.-l. Wan, R.-j. Chen, and S. Wang, "Feature-Selection-Based Ransomware Detection with Machine Learning of Data Analysis," 2018 3rd International Conference on Computer and Communication Systems (ICCCS), pp. 85-88, 2018.

[8] A. El-Kosairy and M. A. Azer, "Intrusion and ransomware detection system," in 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), 2018.

[9] N. Hampton, Z. Baig, and S. Zeadally, "Ransomware behavioural analysis on windows platforms," J. Inf. Secur. Appl., 2018.

[10] N. B. Harikrishnan and K. P. Soman, "Detecting Ransomware using GURLS," Second Int. Conf. Adv. Electron. Comput. Commun., pp. 1–6, 2018.

[11] Z.-G. Chen, H.-S. Kang, S.-N. Yin, and S.-R. Kim, "Automatic Ransomware Detection and Analysis Based on Dynamic API Calls Flow Graph," in Proceedings of the International Conference on Research in Adaptive and Convergent Systems RACS '17, 2017.

[12] Y. Takeuchi, K. Sakai, and S. Fukumoto, "Detecting Ransomware using Support Vector Machines," the 47th International Conference on Parallel Processing Companion - ICPP '18, 2018.

Journal of Electronical & Cyber Defence
9 Vol. 8, No. 4, 2021, Serial No. 32

Improvement in the Ransomwares Detection Method With New API Calls Features

H. R. Javaheri, H. Akbari*, E. Shaghaghi
*Imam Hossein Comprehensive University

(Received: 10/12/2019, Accepted: 01/02/2020)

ABSTRACT

In recent years, the tendency for ransomware-based cyberattacks has increased dramatically. One of the defensive methods is the behavioral detection of the ransomware by system functions. Literature review and related studies and investigations in this field show that these researches are not optimum concerning the accuracy and speed of ransomware detection. Because all datasets used in these studies are limited in scope, they have shortcomings such as high false positive or false negative rates and even high indiscriminate rates. Another drawback of these schemes is the failure to expedite the debate on extortion ransom. Therefore, in this study, the first step is to generate an initial dataset with 126 attributes containing all types of ransomware families. Then, by performing 4-step experiments and tests and applying a feature selection algorithm, this initial set is processed and optimized and reduced to a dataset with 67 attributes without loss of detection precision. In the final step, by providing an optimal and so-called lightweight dataset, the best classification model for the detection of ransomware is obtained being capable of identifying ransomwares with an optimum precision rate of 95.11 in 0.21 seconds, a false positive rate of 0.047 and a true positive rate of 0.951 by using a random forest classification algorithm (using 10-part cross-validation method).

Keywords: Ransomware, Feature Selection, API Calls, Detection Accuracy Rate, Classification, Machine Learning, Dataset, Rate of Detection Speed

* Corresponding Author Email: hamidakbari@ihu.ac.ir