

علمی- پژوهشی

بهبود روش شناسایی باج افزارها با استفاده از ویژگی های توابع سیستمی

حمیدرضا جواهری<sup>۱</sup>، حمید اکبری<sup>۲\*</sup>، احسان اله شقاقی<sup>۳</sup>

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 سال های اخیر گرایش حملات سایبری مبتنی بر باج افزارها به شدت افزایش یافته است. یکی از روش های پدافندی، شناسایی رفتاری باج افزارها به وسیله توابع سیستمی است. با مطالعه و بررسی پژوهش های این حوزه دریافتیم پژوهش های مذکور در نرخ دقت و سرعت تشخیص باج افزارها بهینه نمی باشد. به دلیل اینکه جامعه آماری نمونه باج افزارهای مجموعه داده های مورد آزمایش و ارزیابی در این پژوهش ها محدود بوده و همه خانواده های باج افزاری را پوشش نمی دهد، لذا میزان نرخ های تشخیص ارائه شده برای شناسایی تعداد بالای باج افزارها دارای کاستی هایی چون پایین بودن نرخ دقت تشخیص، نرخ بالای مثبت کاذب و حتی بالا بودن نرخ عدم تشخیص هستند. از دیگر کاستی پژوهش های مذکور غفلت از تأثیر نرخ سرعت در تشخیص باج افزارها است. عدم رفع کاستی های مذکور در زمان پیاده سازی این گونه روش های شناسایی، موجب متحمل شدن هزینه های 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 مجموعه داده غنی شامل انواع خانواده باج افزارها و در نسخه های مختلف شده است. در ادامه با انجام آزمون هایی طی ۴ مرحله روی مجموعه داده اولیه با ۱۲۶ ویژگی و برگزیدن الگوریتم انتخاب ویژگی مناسب، اقدام به بهینه سازی آن شده است. در نتیجه مجموعه داده ای بهینه با ۶۷ ویژگی بدون کاهش نرخ دقت تشخیص به دست آمده است. سپس به وسیله این مجموعه داده بهینه و به اصطلاح سبک اقدام به اخذ بهترین مدل دسته بندی برای تشخیص کرده، لذا به وسیله الگوریتم دسته بندی جنگل تصادفی (با استفاده از روش مقابله ای ۱۰ بخشی) موفق به شناسایی باج افزارها با نرخ دقت بهینه ۹۵۶۷/۱۱٪ در مدت زمان ۰/۲۱ ثانیه، نرخ مثبت کاذب ۰/۴۷ و نرخ مثبت صحیح ۰/۹۵۱ شده ایم.

**کلیدواژه ها:** باج افزار، شناسایی رفتاری باج افزارها، انتخاب ویژگی های باج افزارها، باج افزارهای رمزنگار، توابع سیستمی، نرخ دقت و سرعت تشخیص باج افزارها، دسته بندی باج افزارها.

۱- مقدمه

حوزه مورد توجه بوده است [۲]. طبق گزارش شرکت امنیتی سیمانتک حجم گونه های مختلف باج افزار نسبت به سال ۲۰۱۵ در سال ۲۰۱۷<sup>۲</sup> حدود ۴۶٪ افزایش داشته است [۳]. دیگر زمینه بسیار مهم قابل ذکر رشد روزافزون صنعت یا تجارت پول های الکترونیکی است که باعث رشد تولید خانواده ای از بدافزار به نام باج افزار شده است [۴]. باج افزارها دسته ای از بدافزارها هست که با استفاده از مهارت های بداندیش و مخرب دسترسی کاربران به اطلاعات سیستم هایشان جلوگیری می کند [۵].

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

در دنیای امروز فناوری اطلاعات، تحولات زیادی وجود دارد. این تحولات در حوزه سایبری و امنیت آن اتفاق می افتد. در این حوزه روزانه ابزارهای بداندیش<sup>۱</sup> زیادی توسعه و تولید می شوند. در مقابل متخصصان امنیتی این حوزه سعی در شناسایی و جلوگیری از این گونه فعالیت ها دارند. طی گزارش از سوی اتحادیه اروپا باج افزارها با توجه به بهره مندی از کانال ارتباط فرماندهی و کنترل با سرورهای حمله گر به عنوان بات نت دسته بندی شده اند [۱]. باج افزارها معمولاً با قرنطینه کردن سیستم قربانی یا اطلاعات درون آن به وسیله رمزنگاری فایل ها، اطلاعات قربانی و یا قفل کردن سیستم برای رسیدن به اهداف خود تلاش می کند [۲]. ابعاد حوزه جرائم سایبری روزانه در حال رشد است و به طور طبیعی برنامه های بداندیش نیز به عنوان یکی از زمینه های این

\*ایانامه نویسنده مسئول: hamidakbari@ihu.ac.ir

<sup>1</sup> Malicious

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 به دست 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 احصاء شده اقدام کرده‌اند.

نیکلای همپتون و همکارانش، به بررسی رفتاری باج‌افزارهای آلوده‌کننده سیستم عامل 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 باج‌افزارها نموده‌اند.

کریشنان و سومن در مدلی به نام GURLS به شناسایی و دسته‌بندی باج‌افزارها پرداخته‌اند [۱۰]. ویژگی‌های استخراج‌شده در این طرح از تحلیل ایستا و نیز رشته‌های مورد استفاده باج‌افزار به دست 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.

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 بیش از پیش مشخص گردید. مراحل شناسایی باج‌افزارها به دو صورت قابل طرح 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. در روش دوم شناسایی بر اساس تحلیل باج‌افزار و پس از مقیم شدن آن در سیستم قربانی صورت می‌گیرد [۶]. البته این روش‌ها هم‌زمان بر 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 مقیم شدن 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 خاص‌منظوره‌تری چون قفل کردن 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 کم دقت 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 مجموعه داده 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 بهینه کردن روش‌های موجود شناسایی باج‌افزارها خلاء قابل ملاحظه‌ای بود که در روش‌ها و پژوهش‌های موجود در این زمینه

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 هم زمان 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 عمل 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 خاص‌منظوره‌تری چون عملیات قفل کردن سیستم و یا رمزنگاری دارند. این گونه 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 مجموعه داده که شامل جمع‌آوری فایل‌های نمونه باج‌افزار و پاک است پرداخته شد. لازم به ذکر است با توجه به تلاش‌های صورت گرفته به دلیل عدم دسترسی مجموعه داده استاندارد در این حوزه مجبور به تولید مجموعه داده 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 به وسیله 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 مرحله این طرح اقدام به نوشتن پارسر و سپس انتخاب ویژگی کردیم. چون هدف طرح مد نظر

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 فراخوانی شده 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 به وسیله 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 جعبه‌شن مورد تحلیل پویا قرار گرفته‌اند. گزارش‌های حاصل از تحلیل پویای این باج‌افزارها از جعبه‌شن استخراج گردیده است. بر اساس ویژگی‌های خاص توابع فراخوانی شده توسط باج‌افزارها، فعالیت‌های مخرب و مشکوک شناسایی شده است و بر اساس ویژگی‌های استخراج شده 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 به وسیله 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 انجام شده در این حوزه و بررسی‌های صورت گرفته روی این پژوهش‌ها و نیز نتایج به دست آمده حاصل از آن، می‌توان کاستی‌های این پژوهش‌ها را در ۳ مورد بیان کرد:

- نبود یک مجموعه داده 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، به دلیل 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 به عنوان 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.

پاک تهیه کرد. لذا فرایند تولید مجموعه داده را به تفصیل در ادامه شرح داده ایم.

### ۳-۱-۱- تولید نمونه فایل های اجرایی

برای تهیه مجموعه نمونه فایل های باج افزار سعی کردیم از سطح اینترنت و سایت هایی چون ویروس ساین<sup>۱</sup> یا مل شیر<sup>۲</sup> و یا ویروس توتال<sup>۳</sup> و دیگر منابع مرتبط تهیه کنیم. در ادامه و با تحقیقات اقدام به نصب و راه اندازی ظرف غسل شدیم تا از طریق آن بتوانیم حملات شامل فایل های باج افزار را به دام اندازیم. در نهایت توانستیم به تعداد ۳۱۱ عدد نمونه فایل باج افزار 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 فایل های مجموعه داده مربوطه به فایل های اجرایی پاک و بی خطر تعلق دارد. این دسته فایل نیز که به تعداد ۳۹۰ عدد می باشد، نیز از سطح اینترنت جمع آوری شده است و شامل انواع دسته بندی های فایل اجرایی می باشد. تعداد ۶۷ نمونه فایل پاک نیز برای مرحله ارزیابی در نظر گرفته شده است. پس از فراهم سازی این مجموعه فایل ها، حال نیاز است که رفتارهای آن ها پس از اجرا تحت نظارت و عملیات مانیتورینگ قرار بگیرد.

### ۳-۱-۲- نظارت بر رفتار اجرای فایل ها

پس از تهیه مجموعه نمونه فایل های اجرایی شامل انواع خانواده باج افزارها و انواع دسته بندی فایل های اجرایی بی خطر، حال نسبت به اجرا و نظارت بر رفتارهای آن ها اقدام می کنیم. در ادامه ابتدا نحوه نظارت بر رفتار این فایل ها پس از اجرا که شامل تنظیمات محیط، اقدامات امن سازی محیط، نحوه اجرای فایل ها و استخراج گزارش ها است را تشریح می کنیم.

### ۳-۲- محیط اجرای فایل ها

همان طور که در پیش تر اشاره شد رویکرد این پژوهش مبتنی بر رفتار است. تحلیل مبتنی بر رفتار یا به نوعی تحلیل پویا چیزی جز اجرای برنامه ها نیست. لذا با توجه به ویژگی مخرب بودن و خرابه کارانه باج افزارها نباید آن ها را در محیط عادی و بدون حفاظت اجرا کرد؛ زیرا انجام این امر موجب تخریب نرم/سخت افزاری سیستم مربوطه، سرقت و از بین رفتن داده ها و دیگر موارد احتمالی گردد؛ بنابراین برای جلوگیری از تحقق خسارات احتمالی نیاز به محیط محافظت شده برای اجرای نمونه فایل های باج افزار با شرایط یکسان است. برای این منظور از بستر مجازی سازی Esxi نسخه ۵.۵ روی سرور استفاده نمودیم. سپس سامانه ثبت رخداد را روی یک میزبان با سیستم عامل 12.04 Ubuntu نصب کرده و برای اجرای فایل ها ۴ ماشین مجازی

شناسایی از طریق توابع سیستمی فراخوانی شده است لذا اقدام به نوشتن پارسی می کنیم که با تفسیر کردن فایل های Json حاوی رفتارهای حین اجرای مجموعه فایل های 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 پیش پردازش مورد نیاز است تا در آن بین تمامی توابع فراخوانی شده، اقدام به شناسایی توابعی کرده ایم که در اکثر باج افزارها مورد استفاده قرار گرفته است. این توابع برای تشخیص فایل های باج افزارها از بالاترین اولویت برخوردارند زیرا مختص باج افزارها هستند. این فهرست توابع در مراحل بعد برای انتخاب ویژگی های شناسایی کاربرد تأثیرگذاری خواهد داشت. در گام بعدی از این مرحله نسبت به انتخاب توابع مناسب به عنوان انتخاب ویژگی ها اقدام می کنیم. در این بخش روش های انتخاب ویژگی مورد مطالعه و بررسی قرار گرفته تا برای انتخاب بهترین توابع سیستمی جهت شناسایی دقیق تر باج افزارها مورد استفاده قرار گیرد. روش های انتخاب خصیصه جهت انتخاب زیرمجموعه ای کوچک تر از مجموعه اصلی خصیصه ها مد نظر قرار می گیرد.

در مرحله سوم و در آخرین مرحله این طرح نسبت به اخذ مدلی برای دسته بندی فایل های خوش خیم و باج افزارها از طریق یادگیری ماشین اقدام شد تا با اخذ بهترین مدل طبقه بندی روی ویژگی های اخذ شده در مرحله قبل، نسبت به شناسایی باج افزارها عمل شود. در این مرحله نسبت به مطالعه و بررسی الگوریتم های دسته بندی یادگیری ماشین اقدام گردید.

همان طور که پیش تر بیان شد، نرخ تولید باج افزارها در سال ها و حتی ماه های گذشته به طور فزاینده ای رو به رشد هست. به نوعی می توان گفت حمله کنندگان سایبری میل و رغبت مضاعفی برای استفاده از باج افزارها در حملات خود داشته اند. در اثر این حملات هزینه های گزافی به قربانیان تحمیل شده است. هزینه هایی چون مالباختگی و مهم تر از همه سرقت، افشاء و یا پاک شدن اطلاعات مهم قربانیان از جمله این خسارات است. لذا هدف اصلی این پژوهش بهینه سازی روش های شناسایی پیشین ارائه شده است. این بهینه سازی مشتمل بر حالت بهینه در نرخ دقت تشخیص و سرعت آن است. چرا که هر چه سرعت و دقت تشخیص بالاتر باشد، در حجم های بالای تعداد باج افزارها نرخ شناسایی به مراتب افزایش می یابد.

### ۳-۱- تولید مجموعه داده

در گام ابتدایی این طرح نیاز است تا مجموعه ای از نمونه فایل های اجرایی شامل دسته ای از انواع باج افزارها و دسته ای از انواع فایل

<sup>1</sup> Virus Sign

<sup>2</sup> Malshare

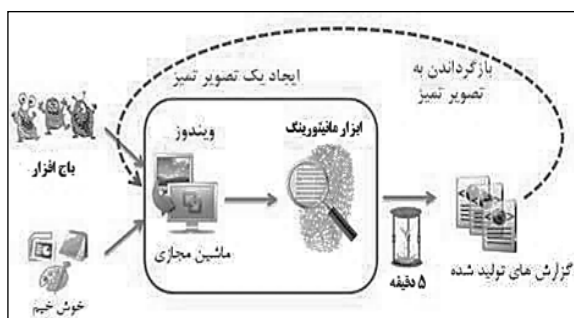
<sup>3</sup> Virus Total

سپس منتهی به ایجاد گزارش نهایی نشوند حذف می شوند. معیار ۵ دقیق با توجه به زیرساخت مورد استفاده و آزمایش های 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 باج افزارهایی با این ویژگی زمان بندی اجرا در فایل تجمیع شده نهایی ویژگی های رفتاری قرار نگرفته و ویژگی های اخذ شده 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 فایل های نمونه است. روند اجرای فایل ها بدین صورت است که ابتدا فایل اجرایی به عنوان ورودی به جعبه شن داده می شود و جعبه شن با توجه به درخواست و یا نیاز فایل ورودی محیط و سیستم عامل مربوطه را فراخوانی و اجرا می کرد. سپس قبل از اجرای فایل جعبه شن یک فایل پشتیبان جدید برای این فایل ورودی ایجاد می کرد. فایل ورودی را در سیستم عامل مد نظر اجرا کرده، نسبت به سیر مراحل نظارت اقدام می کرد.

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 می گرداند تا شرایط برای اجرای فایل بعدی مهیا و یکسان باشد. در برخی موارد فایل ورودی برای اجرا نیاز به یکسری فایل های سیستمی وابسته داشت که به صورت دستی این فایل ها 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 داده شده 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 می گیرد.

### ۳-۷- ابزار پارسر

پس از اجرای تمامی نمونه فایل ها و اخذ تمامی گزارش های رفتاری حاصل از آن ها حال با توجه به اینکه موضوع این پژوهش

با سیستم عامل های ویندوز xp و ویندوز ۷ در نسخه های ۳۲ و ۶۴ بیتی راه اندازی نمودیم تا این مجموعه ابزارها برای اجرای نمونه فایل ها مهیا گردد و فرایند ثبت ویژگی ها آغاز گردد.

### ۳-۳- ابزارهای ثبت رفتار

پس از مهیاسازی محیط محافظت شده برای اجرای نمونه فایل ها حال باید ابزارهای مورد نیاز برای اجرای عملیات اخذ ویژگی های رفتاری حین اجرای فایل ها را نصب و راه اندازی نماییم. برای این منظور اقدام به نصب و راه اندازی جعبه شن کوکو نسخه ۲،۰ کردیم. این جعبه شن اقدام به اجرای فایل های نمونه پاک و باج افزار کرده و تمامی ویژگی ها را از زمان اجرای آن ها تا پس از اجرا ثبت می کند. سپس تمام ویژگی های اخذ شده را در قالب فایل Json ارائه می دهد. خروجی های این جعبه شن بسته به نوع مأموریت و فعالیت فایل اجرایی دارای حجم فایل و تعداد ویژگی های متفاوت می باشد.

### ۳-۴- مقاوم سازی جعبه شن

یک نکته مهم و حائز اهمیت در این مرحله مقاوم سازی جعبه شن در برابر رفتارهای بدخواهانه باج افزارها است. برخی باج افزارها اقدام به تشخیص محیط های مجازی کرده و با توجه به شواهدی چون سیستم فایل های بسترهای مجازی سازی در مسیر رجیستری یا نصب و راه اندازی برخی خواص و ویژگی های این سیستم ها توان شناسایی محیط های مجازی را دارند. لذا نیاز است تا نسبت به جلوگیری از شناسایی توسط باج افزارها اقدامات مقاوم سازی را مد نظر قرار دهیم. به عنوان مثال چون ماهیت فعالیت باج افزارها بر مبنای عملیات خواندن و نوشتن است لذا این گونه 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 می شد. لذا اقدام به پیشگیری و اعمال روش های عدم شناسایی محیط اجرا توسط باج افزارها گشته و در برخی موارد مجبور شدیم جهت اطمینان از صحت مرحله اخذ ویژگی ها از این گونه باج افزارها در مجموعه داده خود استفاده نکنیم.

### ۳-۵- معیار مدت زمان محدود برای اجرای فایل

برخی باج افزارها رفتار خود را در زمان بندی خاصی بروز می دهند و در کد برنامه ای آن ها بر حسب نوع مأموریت و محیط مد نظر برای اجرای خود تایمرهای زمان بندی به کار گرفته شده است. در برخی موارد ممکن است یا باج افزار رفتار خود را پس از ۲۴ ساعت، چند روز، چند هفته و یا حتی چندین ماه و بیشتر بروز دهد. برای این گونه موارد زمان بندی ۵ دقیقه ای از زمان اجرا ملاک قرار داده شده است. بدین صورت که فایل هایی که پس از ۵ دقیقه اجرا و

شبهات 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 تفکیک‌کننده در مجموعه داده و یا ویژگی‌های آن میسر می‌شود. پس از کشف الگوهای مناسب و

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 فایل csv 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 نامرتب 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

بخش دوم مجموعه داده شامل فایل های پاک قابل حمل است. این بخش از نمونه فایل ها نیز از دسته بندی های مختلف برنامه ها تهیه شده است. از جمله این دسته بندی ها می توان به چندرسانه ای، اسنادی، امنیتی، گرافیکی، نظارتی، بازی و فایل های سیستمی اشاره کرد. این فایل ها اغلب به دلیل قابل حمل بودن دارای حجم پایین می باشد. در نهایت تعداد ۳۹۰ فایل پاک برای مجموعه داده ابتدایی این طرح جمع آوری 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 مجموعه داده ابتدایی و نهایی به همراه داده های آماری آن ها نشان داده شده است.

جدول (۲): میزان فراوانی نمونه فایل های باج افزار نسبت به خانواده

باج افزارها

خانواده های باج افزار	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 فایل های بی خطر جمع آوری شده نسبت به دسته بندی های متعدد آن را نشان می دهد.

جدول (۳): فراوانی نمونه فایل های بی خطر نسبت به حوزه های مختلف

دسته بندی فایل های پاک	ارزیابی	آموزش
چندرسانه ای	۸	۳۹
اسنادی	۱۰	۴۹
امنیتی	۸	۴۶
گرافیکی	۷	۳۹
نظارتی	۱۳	۵۷
بازی	۵	۲۹
سیستمی	۱۶	۶۴

در جدول (۳) می توان میزان فراوانی نمونه فایل های پاک قابل حمل و باج افزار را نسبت به کل مجموعه داده در مراحل ارزیابی و آموزش مشاهده نمود.

متناسب با مجموعه داده، این الگوها برای دسته بندی نمونه فایل های ناشناخته مورد استفاده قرار می گیرد. با توجه به اینکه تنها دو دسته یا دو کلاس فایل باج افزار و پاک داریم لذا از الگوریتم های دسته بندی دودویی برای شناسایی آن ها استفاده می کنیم.

هدف اصلی، دسته بندی فایل های پاک از باج افزارها با نرخ دقت بالا است. به همین منظور ورودی این الگوریتم ها فایل ویژگی های بهینه شده در مرحله قبل می باشد. الگوریتم های درخت تصمیم J48، جنگل تصادفی، NaiveBayes، لاجستیک ساده، SMO، کرنل خطی، پایه شعاعی در این پژوهش مورد استفاده قرار گرفته است. این الگوریتم ها به وسیله فرآیند اعتبارسنجی ۱۰ قسمتی برای جلوگیری از احتمال هم پوشانی بیش از حد مدل مورد ارزیابی استفاده شده است.

#### ۴- ارزیابی و تحلیل نتایج

در این بخش به بررسی و صحت سنجی روش پیشنهادی ارائه شده در این پایان نامه خواهیم پرداخت. لذا در ادامه به بررسی نتایج حاصل از آزمایش ورودی های این طرح در قالب جداول و تصاویر آن می پردازیم. جزئیات داده های مورد استفاده، معیارهای ارزیابی، سوالات مطرح شده و پاسخ به آن ها در این فصل شرح داده خواهد شد.

#### ۴-۱- مجموعه داده ها و تنظیمات آزمایش ها

در این پژوهش دو مجموعه داده مورد استفاده قرار گرفته است. مجموعه داده ابتدایی که شامل فایل های اجرایی پاک و باج افزار است. این فایل ها از فضای اینترنت وب سایت هایی چون ویروس ساین یا مل شیر و یا ویروس توتال جمع آوری شده است. در تهیه بخش فایل های باج افزار تلاش بر این بوده است که برای صحت بیشتر نتایج طرح و فراگیر بودن و جامع بودن آن، از گونه های مختلف باج افزارها شامل خانواده متعدد باج افزارها مانند Cerber, Cryptolocker, Cryptowall, Locky, Sage, wannacry, Torrentlocker استفاده شده است. به علت کمبود نمونه فایل های باج افزار در فضای اینترنت و نیاز به تأمین این فایل ها و نیز عدم دسترسی به خرید و دانلود بانک بدافزارها از شرکت های امنیتی سایبری به علت تحریم ها، بنا به تجربه قبلی اقدام به نصب و راه اندازی ظرف غسل یا تله سایبری<sup>۱</sup> در مسیرهای اینترنتی آلوده شده و توسط آن برخی از فایل های مجموعه داده را تهیه کردیم. در نهایت توانستیم به تعداد ۳۱۱ عدد نمونه فایل باج افزار از انواع خانواده آن تهیه کنیم.

<sup>1</sup> Honeypot

## ۴-۲- مرحله انتخاب ویژگی

محاسبه کنیم. لازم به ذکر 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 به دست آمده مهارت gain ratio به وسیله 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 حاصل شده 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 مجموعه داده 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 به دست آمده را به وسیله الگوریتم‌های یادگیری ماشین مد نظر قرار داده و محاسبه کنیم. لازم به ذکر 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.

در این مرحله که مهم‌ترین بخش این طرح می‌باشد، باید عملیات انتخاب ویژگی صورت گیرد. این انتخاب باید به گونه‌ای باشد تا ویژگی‌های منتخب منجر به ایجاد بهینه‌سازی در سرعت و دقت نرخ تشخیص با افزایش آنها 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 مجموعه داده 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 به کار برده 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 مجموعه داده آن ۱۲۵ عدد است لذا ویژگی‌های فعلی اخذ شده مورد استفاده در این پایان‌نامه ۲۳۲ عدد است و باید نسبت به بهینه‌سازی تشخیص آن که 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

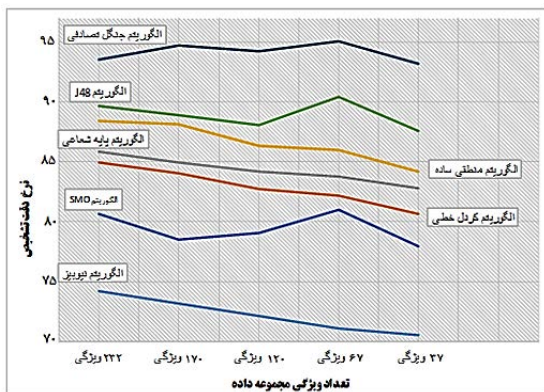
در ابتدای گام بهینه‌سازی از الگوریتم gain ratio برای انتخاب ویژگی مورد استفاده قرار دادیم. این الگوریتم نسبت اطلاعات به دست آمده از ویژگی‌ها به اطلاعات ذاتی خود ویژگی را محاسبه می‌کند. پس از اعمال الگوریتم انتخاب ویژگی GainRatio روی مجموعه داده تعداد ۱۱۷ ویژگی از میان ۲۳۲ ویژگی بر اساس این الگوریتم انتخاب 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 به دست آمده را به وسیله الگوریتم‌های یادگیری ماشین مد نظر قرار داده و



جدول (۵): نتایج حاصل شده از اعمال الگوریتم انتخاب ویژگی

الگوریتم دسته بندی	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

ارزیابی نموده تا میزان این نرخ کاهش نیابد. با تکرار این کار به مرحله‌ای می‌رسیم که با کاهش تعداد ویژگی‌های مجموعه داده نرخ دقت تشخیص نیز کاهش می‌آید و آنجا همان آستانه تحمل نرخ دقت و سرعت تشخیص این مجموعه داده یا به عبارتی نقطه عطف و تلاقی این نرخ می‌باشد. در جداول زیر نتایج به دست آمده از ارزیابی مجموعه داده در هر مرحله به تفکیک الگوریتم‌های دسته بندی به تصویر کشیده شده است. در جدول (۴) نتایج ارزیابی بر اساس الگوریتم دسته بندی 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 دسته بندی نقطه نزولی نرخ دقت تشخیص، نقطه‌ای است که ویژگی‌های مجموعه داده از عدد ۶۷ روند کاهشی را آغاز می‌کند. این اثبات کننده این است که تعداد بهینه ویژگی برای مجموعه داده تعداد ۶۷ است. نکته دیگر حائز اهمیت این است که با توجه به نمودار مشاهده می‌شود که الگوریتم‌های دسته بندی درخت تصمیم یعنی جنگل تصادفی و J48 نسبت به دیگر الگوریتم‌ها از تناسب بهتری با مجموعه داده 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 دسته بندی مذکور می‌پردازیم. در ابتدا به نمودار (۲) توجه کنید.

همان طور که در نمودار (۲) قابل مشاهده است، به طور طبیعی زمان تشخیص در اکثر الگوریتم‌ها با کاستن از ویژگی‌ها رو به کاهش می‌باشد. این نرخ نزولی سرعت تشخیص در الگوریتم‌های دسته بندی حوزه رگرسیون و نیز الگوریتم J48 با سرعت کمتری می‌باشد. دلیل آن هم این است که این دسته الگوریتم‌های دسته بندی دارای ساختاری سبک و چابک تر و در نتیجه سریع تر 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

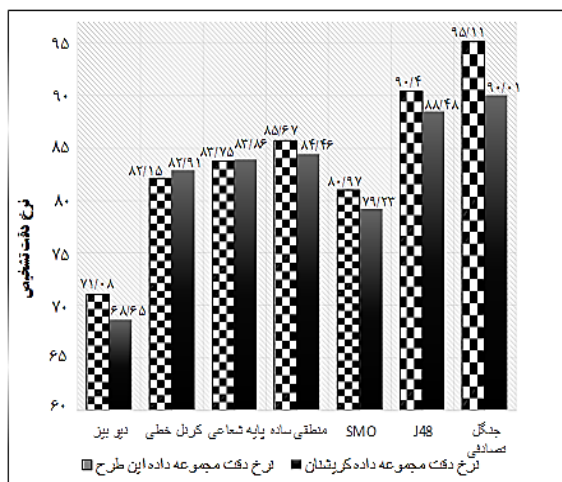
اگر نتایج حاصل شده در جدول (۵) را مورد ارزیابی قرار دهیم، مشاهده می‌کنیم که نرخ‌های به دست آمده دارای ویژگی‌های خاص به خود هستند و این بر اساس نوع الگوریتم دسته بندی می‌باشد. برخی از الگوریتم‌های دسته بندی مانند بیز ساده بسیار سبک بوده و دارای سرعت بالایی است که در اینجا زمان تشخیص آن ۰/۰۸ ثانیه برآورد 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 پیاده سازی دارند زمان بالاتری می‌طلبند که در اینجا ۱/۲ ثانیه می‌باشد.

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

همان طور که در قسمت‌های قبلی شرح داده شد دو الگوریتم انتخاب ویژگی برای بهینه سازی بیشتر روی مجموعه داده اعمال شده و ویژگی‌های استخراج شده بر مبنای این دو الگوریتم نمایش داده شد. در ادامه اقدام به اعمال الگوریتم‌های دسته بندی روی ویژگی‌های به دست آمده کردیم و نرخ‌های معیارهای تشخیص را به دست آوردیم که در قالب جداول ارائه گردید.

در ادامه روند بهینه سازی مجموعه داده اقدام به انتخاب ویژگی‌ها در چند مرحله به وسیله 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 مرتب سازی شده از تعداد فراخوانی توابع این مجموعه داده 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 مجموعه داده کاسه می‌شود و توقع می‌رود زمان تشخیص کاهش یافته باشد. از طرفی باید در هر مرحله نرخ دقت در تشخیص را

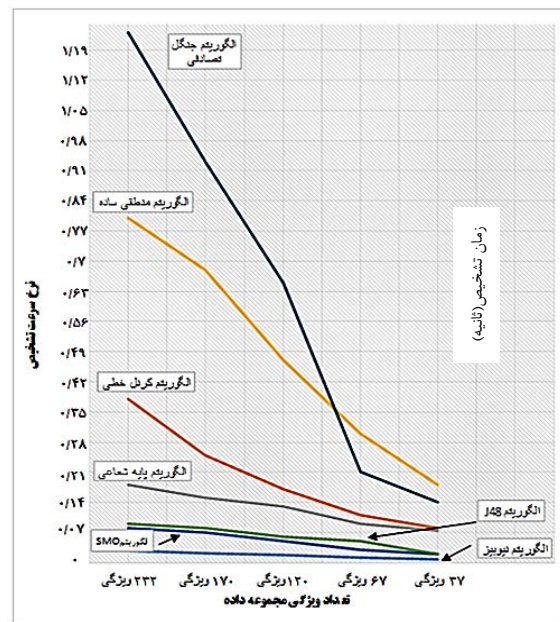
با توجه با نتایج حاصل شده از اعمال انواع الگوریتم‌های دسته‌بندی روی مجموعه داده می‌توان دریافت که الگوریتم‌های حوزه درخت تصمیم از آمار نرخ تشخیص بهتری نسبت به دیگر الگوریتم‌ها برخوردارند. با ارزیابی دقیق‌تر نسبت به نتایج حاصل شده و بر مبنای معیارهایی چون نرخ دقت تشخیص، نرخ مثبت کاذب، نرخ مثبت صحیح و 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 مجموعه داده با ۶۷ ویژگی الگوریتم بهینه دسته‌بندی است. لازم به ذکر 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 اعلام شده در طرح و محاسبه زمان تشخیص بر مبنای الگوریتم دسته‌بندی مربوطه) روی مجموعه داده با ۱۲۰ ویژگی بوده است. در بهترین حالت نرخ دقت تشخیص مجموعه داده طرح کریشنان و سومن، که در این پژوهش آن را بر مبنای ویژگی‌های مجموعه داده مندرج در آن طرح و نیز بر اساس الگوریتم جنگل تصادفی ارزیابی کردیم؛ نرخ دقت معادل ۹۰/۰۱٪، نرخ 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.

تندی قابل ملاحظه است و این نشان می‌دهد با اعمال عملیات حذف ویژگی‌های چندمرحله‌ای روی مجموعه داده بر اساس الگوریتم جنگل تصادفی شاهد افزایش نرخ سرعت و در نتیجه تسریع در عملیات شناسایی خواهیم بود. پس اگر کمیت مجموعه داده ما چندین برابر مجموعه داده فعلی باشد با اعمال عملیات بهینه‌سازی ویژگی‌ها روی مجموعه داده 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 به دست آمده در پنج مرحله آزمایش روی مجموعه داده برای استخراج نرخ بهینه سرعت و دقت در تشخیص به این نتیجه می‌رسیم که تا رسیدن به مجموعه داده 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 مجموعه داده برای کشف نرخ بهینه دقت و سرعت در تشخیص، مجموعه داده با ۶۷ ویژگی است.

#### ۳-۴- تحلیل و ارزیابی مرحله انتخاب الگوریتم دسته‌بندی

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

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 مجموعه داده به این مهم پرداختیم. آزمایش‌ها نشان می‌دهد که صرف نظر از مجموعه داده، ویژگی‌های تولید شده قابلیت تمایز بالایی دارند. این مهم به وسیله در نظر گرفتن دو مجموعه داده ابتدایی و نهایی و نیز نوع جدید از باج افزارها به آن افزوده شده بود، مورد بررسی قرار گرفت. طبق نتایج آزمایش‌ها با وجود مشاهده رفتارهای جدید در قالب افزوده شدن فایل‌های باج افزار جدید در مجموعه داده نهایی، ویژگی‌های اخذ شده در مجموعه داده اولیه که مدل شده اند قادر به شناسایی باج افزارهای جدید هستند. همچنین طی نتایج آزمایش‌ها، با به روزرسانی ویژگی‌های جدید و مدل سازی آن‌ها دقت تشخیص افزایش میابد. به روزرسانی ویژگی‌ها به دلیل مرور زمان و مشاهده رفتارهای ناشناخته باج افزارها امری اجتناب ناپذیر و منطقی است.

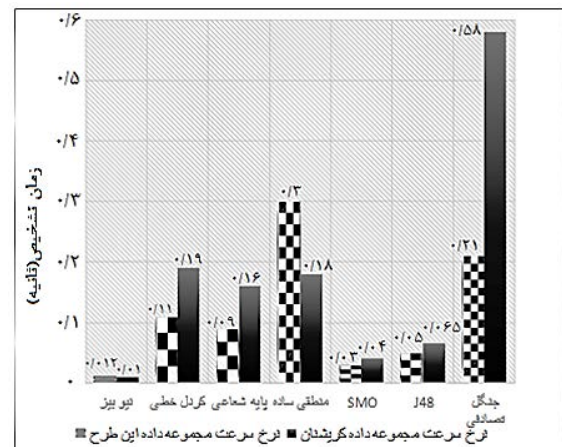
در این طرح در چند مرحله و پس از مطالعه و ارزیابی ابتکارات و نوآوری‌های مورد نیاز به شرح زیر صورت گرفت.

- ایجاد یک مجموعه داده مستقل به وسیله انواع خانواده باج افزارها 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 باج افزارها و استفاده از مشورت متخصصین و تحلیلگران بدافزار.

• استخراج مجموعه داده بهینه از نظر نرخ دقت و سرعت تشخیص به وسیله 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 مجموعه داده می‌گردد.

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

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



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

همان‌طور که در نمودار (۴) مشاهده می‌شود، زمان تشخیص در طرح کیشندان 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 کیشندان 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 کیشندان 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 کیشندان و سومین دارای نرخ تشخیص بهینه می‌باشد.

## ۵- نتیجه‌گیری

در این پژوهش یک روش بهینه کاربردی برای شناسایی پویای باج افزارها ارائه شد. ویژگی‌های استفاده شده 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 به دلیل استفاده از نمونه‌های متنوع فایل‌های مربوطه بوده است. در این طرح نشان دادیم که توابع فراخوانی شده استخراج شده 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

- [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.

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](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>.

## 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