تاریخ دریافت مقاله: ۹۵/۰۷/۰۵
تاریخ پذیرش مقاله: ۹۵/۱۰/۲۴

روشی جدید جهت شناسایی بدافزارهای فراریخت با تحلیل ایستای ثبات‌ها و کدهای عملیاتی

هادی گلباغی
دانشجوی کارشناسی ارشد دانشکده مهندسی و علوم رایانه ـ دانشگاه شهید بهشتی ـ تهران ـ ایران
پست الکترونیکی: h.golbaghi@mail.sbu.ac.ir

مجتبی وحیدی اصل*
استادیار دانشکده مهندسی و علوم رایانه ـ دانشگاه شهید بهشتی ـ تهران ـ ایران
پست الکترونیکی: mo_vahidi@sbu.ac.ir

علیرضا خلیلیان
دانشجوی دکتری دانشکده مهندسی رایانه ـ دانشگاه اصفهان ـ اصفهان ـ ایران
پست الکترونیکی: khalilian@eng.ui.ac.ir

چکیده

شناسایی بدافزارهای فراریخت مسئله دشواری است، زیرا در هر انتشار ساختار کد تغییر می‌یابد درحالی‌که عملکرد و رفتار بدافزار ثابت می‌ماند. تاکنون روش‌های مختلفی برای شناسایی بدافزارهای فراریخت پیشنهاد شده‌اند. گاه روش‌ها با جایگزین کردن دستورات مشابه یا درج کد زائد در بدنه بدافزار دچار شکست می‌شوند و در برخی موارد سربار محاسباتی بالا، دقت شناسایی کم و کارایی ضعیف روش‌ها، آن‌ها را دچار چالش می‌کند. در این مقاله روشی جدید برای شناسایی بدافزارهای فراریخت پیشنهاد می‌شود که با تحلیل ایستای کدهای عملیاتی و ثبات‌های مورد استفاده کار می‌کند. این ایده می‌تواند اساس تمایز بین خانواده‌های مختلف بدافزارها باشد. به منظور ارزیابی روش پیشنهادی، آزمایش‌هایی بر روی ۴۵۰ فایل متشکل از فایل‌های سالم و بدافزار فراریخت از ویروس‌های خانواده‌های G2, MP-CGEN, NGVCK, VLC و کرم‌های خانواده MWOR انجام شد. نتایج آزمایش‌ها نشان می‌دهد که روش پیشنهادی در چهار الگوریتم دسته‌بندی و در تمامی خانواده‌های بدافزارها به دقت شناسایی ۱۰۰ درصدی دست یافته است که با معیار ROC اندازه‌گیری شده است.

واژه‌های کلیدی: بدافزار، فراریختی، مبهم‌سازی کد، موتورهای تکثیر، تحلیل ایستا

* نویسندهٔ مسئول

۱- مقدمه

بر مبنای گزارش‌های ســالیان اخیــر بدافزارها به‌طور روزافزونی تهاجمی‌تر شده‌اند. تحلیلگران شرکت سمانتک[1] نشــان داده‌اند که بدافزاری به اســم Reveton[2] حدود ۵۰۰ هــزار رایانــه را در عرض حــدود ۱۸ روز آلــوده کرده است[۱]. به‌طور متوسط در هر حادثه جرایم سایبری ۱۹۷ دلار از دســت می‌رود[۱]. در ســال ۲۰۱۳ طبق تخمین‌ها حدود ۵۵٦ میلیــون کاربر در سرتاســر جهان یک نمونه از جرایــم ســایبری را تجربــه کرده‌اند. در ســال ۲۰۱۱ مشتری‌ها بیشــتر از ٤۹ هزار گزارش مختلف از تهدیدات خانواده‌هــای مختلــف بدافزارها را به مرکــز حفاظت از بدافزارهای مایکروسافت[3] داده‌اند. تعدادی از این گزارش‌ها تهدیداتــی از جنس خانواده‌های بدافزارهای چندریختی[4] و فراریختی[5] بوده اســت [۱]. در وضعیت کنونی بدافزارهای مختلف رشــد بســیار زیادی کرده‌اند و باعــث اختلالات اساسی در سامانه‌های حساس[6] و حیاتی مانند اقتصادی، مالی، نظامی، پزشکی، سیاسی و غیره شده‌اند [۲]. مزایای اقتصادی بســیاری که در این حــوزه وجود دارد، صنعت بدافزارهای مخرب را به بازاری مستعد برای تولیدکنندگان و نویســندگان بدافزارهــا تبدیــل کرده اســت. از این رو روش‌های جدید و هوشــمند شناســایی هم در بســیاری از مواقــع در حرکت همگام با تولیــد بدافزارهای جدید و شناســایی آن‌ها، عاجز بوده‌اند [۳]. یکی از انواع بدافزارها که هنوز هم چالش‌های عمده‌ای در کشــف آن وجود دارد، بدافزارهای فراریخت هستند. ساختار این بدافزارها در هر تکثیر با اســتفاده از فنون مختلف مبهم‌ســازی کد[7]، تغییر می‌یابد اما عملکرد اصلی آن‌ها حفظ می‌شود و این مسئله، شناسایی آن‌ها را بسیار دشوار و پیچیده می‌کند [۲]. مولد یک بدافزار فراریخت همیشه در تکثیرهای بعدی کد، جهش ایجــاد می‌کند به‌طوری که برای هــر ویروس در هر نفوذ، یک امضای[8] متفاوت ایجاد می‌شود که از شناسایی توسط ضد بدافزارهــا فرار کند [٤]. علاوه بر این، انواع دیگری از بدافزارها مانند اســب تــروا[9] و کرم‌هایی[10] مبتنی بر همین فنــون فراریختی وجود دارند [٤]. در یک دســته‌بندی کلی می‌توان روش‌های کشــف بدافزار را به دو دســته ایستا[11] و پویا[12] تقســیم کرد. روش‌های ایســتا بســیار محبوب و رایجنــد زیرا بدون اجرای بدافزار و از تحلیل ســاختار کد آن، قادرند آن‌ها را کشف نمایند. به‌طور معمول، در روش ایستا برای شناسایی بدافزارها، روش‌ها و ابزار تشخیص با استفاده از روش‌های مبتنی بر امضا به‌طور گسترده‌ای استفاده می‌شود. ساختارهای رشــته‌ای کوتاه از بایت‌ها که منحصربه‌فرد بوده، الگویی از آن‌ها استخراج می‌شود، سپس امضاها در بانک‌های اطلاعاتی نگهداری می‌شوند و عملیات شناســایی مبتنی بر آن‌ها صورت می‌گیرد [۳]. در مقابل، روش‌های پویــا با اجرای بدافزار می‌توانند آن‌ها را کشف کنند که این کار معمولا نیاز به تقلید[13] دارد [٦].

روش‌های متعددی بر مبنای تحلیل ایستای بدافزار کار می‌کنند. یکی از این روش‌ها شناســایی بر اساس خانواده بدافزارها [٦] اســت که این روش فقط محدود به فایل‌های اجرایی قابل حمل می‌باشــد. همچنین در مقاله [۷] بر روی ترتیب دستورات اجرایی و گراف جریان کنترلی در کدهای اســمبلی کار شده که با استفاده از فنون مبهم‌سازی کد که جریان کنترلی دســتورات را تغییــر می‌دهند، روش دچار چالش خواهد شد. روشــی دیگر استفاده از هیستوگرام[14] فراوانــی کدهای عملیاتی [۸] اســت کــه از طریق درج کد زائد و جانشــینی دستورات مشابه دچار شکست می‌شود. روش شــباهت گراف کدهای عملیاتــی [۹] روش موثری اســت اما اگر از روش جانشینی دستورات مشابه استفاده شــود دچار چالش خواهد شد. همچنین در مقاله‌های [٤,۱] کارهایی بر مبنای کدهای عملیاتی در زبان اسمبلی صورت

1-Semantec
2-Trojan.Ransomlock.G
3-https://www.microsoft.com/en-us/security/portal/mmpc/default.aspx
4-Polymorphic Malware
5-Metamorphic
6-Critical Systems
7-Code Obfuscation
8-Signature
9-Trojan Horse
10-Worm
11-Static Analysis
12-Dynamic Analysis
13-Emulation
14-Histogram

گرفته اســت که با درج کد برنامه‌های ســالم و استفاده از جانشینی دستورات مشابه، روش‌ها با افت دقت شناسایی روبرو خواهند شد.

روش‌های شناسایی موجود دارای مشکلات دیگری نیز هستند. برخی از روش‌ها [۱۰،۳] به دلیل سربار محاسباتی بالا، زمانی در مقابل ویروس‌ها موثر عمل می‌کنند که دارای زمان کافی برای شناســایی باشــند و ضد بدافزار و بانک اطلاعاتی آن، به نرخ ســریعی به‌روزرسانی شوند در غیر ایــن صورت تاثیر مثبت آن‌ها کاهــش پیدا می‌کند [۱۲]. به دلیل وجود چالش‌های مطرح شــده، کشــف کامل و قطعی همه انواع و گونه‌های بدافزارهای چندریختی و فراریختی با اعمال روش‌های مختلف و پیچیده جدید مبهم‌سازی، هنوز نیاز به پژوهش دارد و مسئله تحقیقاتی ارزشمندی است.

در ایــن مقاله بــا تکیه بر تفــاوت نرخ تکــرار کدهای عملیاتــی در خانواده‌هــای مختلــف بدافــزار فراریخت و فایل‌های سالم، روش کشــف جدیدی پیشنهاد شده است. این روش با استفاده از چهار معیار استخراج شده حاصل از تحلیل ایســتا، هر فایل ورودی مشکوک را به دو دسته بدافزار یا ســالم دســته‌بندی می‌کند. روش پیشنهادی با تاکیــد روی پالایش و انتخاب صحیــح کدهای عملیاتی در صدد افزایش کارایی اســت. همچنین در شمارش کدهای عملیاتی دســتورات مشابه را نیز همسان در نظر گرفته که در مقابل روش مبهم‌ســازی جانشــینی دستورات مشابه مقاوم است. شــمارش ثبات‌ها به عنوان یک معیار نیز در این روش مدنظر قرار گرفته شده است. به منظور ارزیابی روش پیشنهادی، آزمایش‌هایی بر روی ٤٥٠ فایل متشکل از فایل‌های ســالم و بدافــزار فراریخــت از ویروس‌های خانواده‌هــای G2, MPCGEN, NGVCK, VLC و کرم‌های خانــواده MWOR صورت گرفته اســت. نتایج آزمایش‌ها نشان می‌دهند در کنار دقت شناسایی ۱۰۰ درصدی، عدم افت دقت شناسایی در جانشینی دستورات مشابه، سادگی و سربار کم محاسباتی از مزایای روش پیشنهادی هستند.

ساختار ادامه مقاله به این صورت است: در بخش دوم کارهای مرتبط مورد بررسی قرار می‌گیرد. دربخش سوم روش پیشنهادی تشریح می‌شود. بخش چهارم به ارزیابی و تفسیر نتایج اختصاص دارد. در نهایت بخش پنجم نیز به نتیجه‌گیری وکارهای آینده می‌پردازد.

۲- کارهای مرتبط

کانفورمــا و همــکاران [۱] روشــی برای شناســایی ویروس‌های فراریخت ارائه کرده‌اند که مبتنی بر شناسایی ویژگی‌های خاص از کدهای اســمبلی مانند دستوراتی که محتوای ثبات‌ها را تغییر می‌دهند یا دســتوراتی که کنترل جریان را تغییر می‌دهند، کار می‌کند. آزمایش‌های صورت گرفتــه در این مقالــه موفقیت ۹۷ درصدی در شناســایی بدافزارهای فراریخت را نشان می‌دهد. از مزایای این روش می‌توان به سادگی آن اشاره کرد. همچنین پیاده‌سازی آن برای ضدبدافزارها کار آسانی می‌باشد و بار زیادی برای سیستم ندارد.

رویکرد مهرا و همکاران در مقاله [٦] بر روی شناسایی و طبقه‌بندی بدافزارهای فراریخت بر اســاس خانواده‌های آن‌ها تمرکز دارد. روش پیشــنهادی به این صورت اســت که گراف جریان کنترلی رســم شــده و گــراف فراخوانی API[۱۵] ها ایجاد می‌شــود. این رویکرد هر بدافزار فراریخت را براســاس ویژگی‌های خانواده‌شان که از هیستوگرام و فرمول اندازه‌گیری کای دو، که بر اساس تحلیل پویا است، طبقه‌بنــدی می‌کننــد. در این مقاله، دقــت در الگوریتم‌های مختلــف طبقه‌بندی از ۸۹ تا ۹۹/۱۰ درصد به‌دســت آمده است.

روش پیشــنهادی کانفورما و همــکاران [٤] به معرفی فنون شناســایی مبتنی بــر این فرض که یــک اثر جانبی مشترک بین بسیاری از موتورهای فراریخت وجود دارد، تکیه کرده است. این روش برای حدود ۱۰۰۰ برنامه مورد آزمایش و بررســی قرار داده شــده و بر اساس نتایج آن به‌طــور دقیق ویروس‌هــای فراریخــت و غیرفراریخت را دسته‌بندی کرده است. این روش دارای دقت نزدیک به ۹۹

15-Application programming interface

درصدی در شناسایی است و از معایب آن نیز می‌توان به این مورد اشـــاره کرد که اگر از روش جانشینی دستورات مشابه یا اضافه کردن کد زائد به کد بدافزار استفاده شود، توزیع دستورات تکراری تغییر می‌کند و روش پیشنهادی دچار شکست در شناسایی خواهد شد.

محمد بن خمس و همـــکاران در مقاله [۱۱] ویژگی‌های تغییر داده نشده در ساختار بدافزارهای فراریخت را برای اســـتفاده در فرایند شناسایی بدافزار با استفاده از ماشین بردار پشتیبانی[۱۶] اســـتخراج می‌کند. خصوصیات n-gram به‌طور مستقیم از ساختار دودویی بدافزار استخراج شده که این خصوصیات به عنوان امضا در نظر گرفته می‌شوند. این خصوصیات مقادیر قابل توجهی از تعداد خصوصیات انتخاب n-gram در حالت اصلی را کاهش می‌دهد. این روش ترکیبی از استخراج امضا n-gram و ماشین بردار پشتیبانی اســـت. نتایج ارزیابی‌ها دقتی در حـــدود ۹۹ درصد و نرخ مثبت کاذب پایینی را برای روش پیشنهادی ثبت کرده است.

تمرکـــز اصلی کومار و همکاران در روش پیشـــنهادی [۱۳]، روی تجزیه و تحلیل و مقایسه برخی از محبوب‌ترین ابـــزار تشـــخیص بدافزارهای مخرب اســـت. در این مقاله، نویسندگان برای تشخیص بدافزارهای مخرب یک عملکرد مقایســـه‌ای با ابزار و فنون موجود ارائه داده‌اند. در مقاله مذکور [۱۷] ابزار شناخته شده تشخیص بدافزارهای مخرب و ۲۹ بدافزار به عنوان یک مجموعه داده برای مقایســـه در نظر گرفته شده است.

چـــن و همکاران در مقاله [۳]، جدا از این‌که بر اســـاس محتویات فایل‌های استخراج شده از نمونه فایل‌ها، بدافزار مخرب را شناســـایی می‌کنند، مطالعه و بررسی چگونگی اســـتفاده از گراف مربوط بـــه ارتباطات فایل‌هـــا را برای تشخیص بدافزارهای مخرب مورد توجه قرار داده‌اند و یک الگوریتم انتشـــار جدید بر اساس گراف ساخته شده برای تشـــخیص بدافزارهای مخرب تازه و ناشناخته را پیشنهاد کرده‌اند. آزمایش‌ها روی ٤٦٧٥ فایل متشـــکل از فایل‌های بدافزار فراریخت و ســـالم انجام گرفته است. بر طبق نتایج به‌دست آمده، این روش موثرتر و کاراتر از روش‌های قبلی شناسایی بدافزارها مبتنی بر داده کاوی بوده است. گرچه روش دارای سربار محاسباتی بالا و کارایی کمی می‌باشد.

16- Support Vector Machine

۳- روش پیشنهادی

تحلیل و بررسی روش‌های پیشنهادی در ادبیات و نقاط قوت و ضعف آن‌ها نشان می‌دهد که موتورهای فراریختی همه چیز را در ساختار کد بدافزار تغییر نمی‌دهند. عملیات روی تعداد بســـیاری از ثبات‌ها عوض نمی‌شود؛ بعضی از ثبات‌ها اصلا تغییر نمی‌کنند و بســـیاری از کدهای عملیاتی در بدنه کـــد نیز بی‌تغییر می‌مانند. این بینش در ســـاختار بدافزارهای فراریخت انگیـــزه طراحی روش جدیدی برای کشف بدافزارهای فراریخت است.

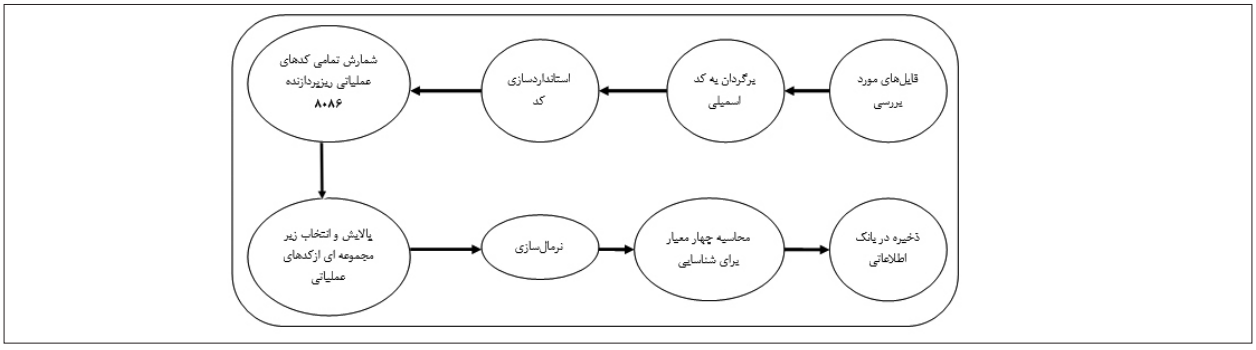
روش پیشنهادی این مقاله، شناسایی بدافزار فراریخت مبتنی بر تحلیل ایســـتا فرض می‌کند که معیار مشـــترکی بین بســـیاری از موتورهای فراریخت وجـــود دارد که در بدنه ویروس‌ها تعداد زیادی از ثبات‌ها، دســـتورالعمل‌ها یا کدهای عملیاتی تکرار می‌شـــوند. ایـــن معیار می‌تواند پایه و اساســـی برای تمایز بین خانواده‌های مختلف بدافزارها باشد و با استفاده از این تمایز می‌توان طبقه‌بندی صحیحی از آن‌هـــا انجام داد. روش پیشـــنهادی شـــامل تحلیل نرخ تکرار دســـتورات برنامه یا همان کدهای عملیاتی و ثبات‌ها است، ســـپس طبق دسته‌بندی نرخ تکرارها، فایل‌های سالم و خانواده‌هـــای مختلف بدافزارهای فراریخت را طبقه‌بندی خواهد کـــرد. مراحل مختلـــف پیش‌پـــردازش اولیه روش پیشنهادی به این صورت است:

پس از پیش پردازش انجام شـــده و محاسبه معیارها، m، n و k رونـــد کلـــی روش پیشـــنهادی برای شناســـایی بدافزارهای فراریخت به‌صورت نشان داده شده در شکل ۱ و شکل ۲ خواهد بود.

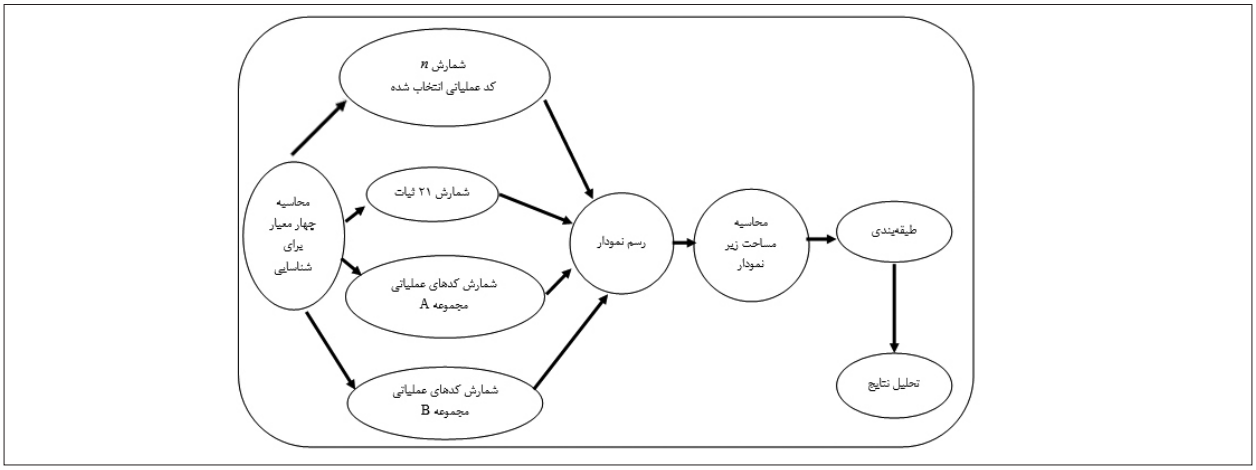
در مقالـــه [٤] از مجموع تمامی کدهـــای عملیاتی، ۱۵٤ مورد را بر مبنای تعیین سطح آستانه نرخ تکرار دو، برای فرایند شناسایی نهایی انتخاب کرده است اما امکان بهبود

الگوریتم: مرحله پیش‌پردازش روش پیشنهادی
ورودی: فایل مورد بررسی
خروجی: کدهای عملیاتی و ثبات‌های شمارش شده
شروع مراحل:
۱- برگردان فایل مورد بررسی به کد اسمبلی
۲- استانداردسازی کد اسمبلی فایل مورد بررسی
۳- شمارش تمامی کدهای عملیاتی ریزپردازنده ۸۰۸٦
٤- پالایش و انتخـــاب n کد عملیاتی و حذف m مورد که نرخ تکرار آن‌ها به حد آستانه تعیین شده k نرسیده باشد
٥- نرمال سازی نرخ تکرارها
٦- محاسبه چهار معیار روش پیشنهادی برای شناسایی و طبقه‌بندی فایل های سالم و خانواده بدافزارهای فراریخت
۷- ذخیره در بانک اطلاعاتی مربوط به شمارش ثبات ها و کدهای عملیاتی خانواده های مختلف
اتمام مرحله پیش‌پردازش

الگوریتم: روند کلی شناسایی بدافزار در روش پیشنهادی
ورودی: فایل مورد بررسی
خروجی: اعمال معیارها و طبقه‌بندی فایل مورد بررســـی به فایل سالم یا بدافزار
شروع مراحل:
۱- تا مرحله ششم مرحله پیش‌پردازش، مجدداً تکرار خواهد شد.
۲- محاسبه چهار معیار به‌صورت زیر:
معیار α) شمارش n کد عملیاتی انتخاب شده طبق حد آستانه k
معیار β) شمارش تمامی ثبات های ریزپردازنده ۸۰۸٦ بجز ثبات فلگ که ۲۱ مورد هستند
معیار γ) شـــمارش کدهای عملیاتی مجموعـــه A که در جدول (۱) نشان داده شده‌اند
معیار δ) شـــمارش کدهای عملیاتی مجموعـــه B که در جدول (۱) نشان داده شده‌اند
۳- رسم نمودار معیارهای محاسبه شده به‌صورت جداگانه
٤- محاسبه مساحت زیر نمودار هر معیار به‌صورت جداگانه
٥- طبقه بندی بر اساس نتایج به دست آمده از معیارها
٦- تحلیل نتایج شناسایی
اتمام مراحل شناسایی



شکل ۱: گام اول نمای کلی روش پیشنهادی مربوط به پیش‌پردازش اولیه



شکل ۲: گام دوم نمای کلی روش پیشنهادی

در انتخاب حد آســـتانه و حذف تعداد بیشـــتری از کدهای عملیاتی کـــه در روند شناســـایی تباعث افزایـــش کارایی روش پیشنهادی خواهد شد. پس از شمارش کل فایل‌های خانواده‌های فراریخت و سالم فقط کدهای عملیاتی پالایش

و انتخاب خواهند شـد که میزان تکرار آن‌ها در کل فایل‌ها بیشـتر از حد آسـتانه k خواهد بود و انتخاب حد آستانه صحیح بر روی کارایی بسیار تاثیرگذار است. برای انتخاب حد آسـتانه و حذف m مورد از کدهـای عملیاتی، یکی از دلایل این اسـت که کدهای عملیاتی که در ساختار کد یک فایل، دارای اهمیت بسـیار کمی هستند و نرخ تکرار آن‌ها بسیار پایین است و اصولا تمایزی بین خانواده‌های مختلف بدافزارها و فایل‌های سالم ایجاد نمی‌کنند باید حذف شوند. پـس از فرایند انتخاب و پالایش و حذف m مورد که تعداد تکرار کمتر از حد آستانه k داشتند، n کد عملیاتی از تمامی کدهای عملیاتی برای فرایند شناسایی نهایی انتخاب شدند که در جدول (۱) نمایش داده شده‌اند. پس از پالایش کدهای عملیاتی، چون نرخ تکرار آن‌ها در فایل‌های مختلف بسیار متفاوت است و این‌که نرخ تکرارها را در یک محدوده قرار داد باید آن‌ها را نرمال‌سـازی کرد. داده‌ها به‌طور مستقیم قابل مقایسه نیستند به این دلیل که تحلیل هر فایل یک بعد از تعداد کل دسـتورات را بیان می‌کنـد در حالی‌که از یک برنامـه با برنامـه دیگر متفاوت خواهد بـود. به‌منظور از بیـن بردن این اختلاف در برنامه‌هـای نمونه و ایجاد داده قابل مقایسه شدن، نیاز به نرمال‌سازی هر جزء از جمعیت تحلیل شده از جنبه‌های از تعداد کل دستورات است. برای نرمال سازی از فرمول (۱) استفاده شده است:

$$\text{(۱) نرمال شده نرخ تکرار هر کد عملیاتی} = \frac{\text{نرخ تکرار هر کد عملیاتی}}{\text{مجموع تکرار تمامی کدهای عملیاتی همان فایل مورد بررسی}}$$

اصلی برای شناسـایی و طبقه‌بندی کردن فایل‌ها محاسبه شـوند. همان‌طور که در جدول (۱) نشـان داده شده است برای هر یـک از چهار معیار، کدهـای عملیاتی مختلف و ثبات‌هایی در نظر گرفته شده‌اند که در زیر به تفکیک مورد بحث قرار می‌گیرند:

معیار α: برای این معیار کل n کد عملیاتی انتخاب شده مدنظر قرار خواهند گرفت.

معیار β: برای این معیار ۲۱ ثبات در نظرگرفته شده‌اند و هر یک در تمامی فایل‌ها شمارش شده و محاسبه می‌شوند.

معیار γ: این معیار به کدهای عملیاتی (کدهای عملیاتی مجموعـه A) می‌پردازد که با توجه بـه نرخ تکرار آن‌ها و تحلیل‌هایـی که بعد از شـمارش تمامـی فایل‌ها صورت گرفته است نرخ تکرار بالایی در فایل‌های سالم دارند و در فایل‌های بدافزار یا اصلا تکرار نشده‌اند و یا تعداد زیر سه بار تکرار را دارند.

معیار δ: این معیار به کدهای عملیاتی (کدهای عملیاتی مجموعـه B) می‌پردازد که با توجه بـه نرخ تکرار آن‌ها و تحلیل‌هایی که بعد از شمارش تمامی فایل ها صورت گرفته اسـت نرخ تکـرار بالایی در فایل‌های بدافـزار دارند و در فایل‌های سـالم یا اصلا تکرار نشده‌اند و یا تعداد زیر سه بار تکرار را دارند.

سـپس نمودار خطی تکرار کدهای عملیاتی که نمودار Xها مربوط به ثبات‌ها یا کدهای عملیاتی و نمودار Yها به تعداد تکرار هر ثبات یا کد عملیاتی اختصاص دارد، رسـم خواهد شد. برای محاسبه چهار معیار، مساحت زیر نمودار مربوط به تکـرار همان کدهای عملیاتی و ثبات‌های جدول (۱) محاسبه می‌شود. سپس برای هر فایل این اعداد در بانک اطلاعاتی ذخیره شـده تا این‌که همه فایل‌ها مورد شمارش قرار گیرند. پس از شـمارش کلیه فایل‌ها و محاسبه تمامی معیارها، کلیه اطلاعات در بانک اطلاعاتی ذخیره می‌شود تا مورد تحلیل و طبقه‌بندی قرار گیرد.

در ارتباط با گسترش روش پیشنهادی به پردازنده‌های امروزی نیـز می‌تـوان در مرحله پیش‌پـردازش کدهای عملیاتی پردازنده‌های جدیدتر را نیز به لیسـت شـمارش اضافه کرد و به این شـکل روش بـه پردازنده‌های جدید نیز بسـط داده خواهد شد. همچنین با توجه به این‌که برای شناسایی بدافزار، برگردان کد وجود دارد و روال همیشگی همه انواع بدافزارها اسـت و چون مرحله پیش‌پردازش یک بار انجام می‌گیرد هزینه محاسـباتی آن سرشکن خواهد شد.

جدول ۱: معیارهای شناسایی و ارزیابی در روش پیشنهادی همراه با کدهای عملیاتی و ثبات‌های انتخاب شده

معیارها / کدها	معیار α :n کد عملیاتی انتخاب شده	معیارβ: ۲۱ ثبات	معیارγ: کدهای عملیاتی مجموعه A	معیارδ: کدهای عملیاتی مجموعه B
کدهای عملیاتی یا ثبات‌ها	ADC,ADD,AND,ARPL,BT,CALL,CDQ,CLC,CLD CLI,CMP,CWD,DEC,DIV,HLT,IDIV,IMUL,IN INC,INS,INT,JA,JNBE,JAE,JNB,JB,JNAE JBE,JNA,JC,JCXZ,JECXZ,JE,JZ,JG,JNLE JGE,JNL,JL,JNGE,JLE,JNG,JMP,JNC,JNE JNZ,JNS,JO,JS,LDS,LEA,LEAVE,LES,LOCK LOOP,LOOPNZ,LOOPNE,MOV,MOVSX,MOVZX NEG,NOP,NOT,OR,OUT,POP,POPA,POPAD PUSH,PUSHA,PUSHAD,RCL,RCR,REP,REPE REPZ,REPNE,REPNZ,RET,RETF,ROL,ROR SAHF,SAL,SHL,SAR,SBB,SCAS,SETB SETNAE,SETE,SETZ,SETNE,SETNZ,SHR STC, STD,STI,STOS,SUB,TEST,XCHG,XOR	AX,BX CX,DX AH,AL BH,BL CH,CL DH,DL CS,DS ES,SS SI,DI SP,BP IP	NOP,MOVZX JLE – JNG,LEAVE JBE – JNA JG - JNLE SETNZ - SETNE CLD,INT SETZ - SETE JGE - JNL JL – JNGE,ROL,JS REPE - REPZ JNB - JAE MOVSX,SAHF SAR,JNS,CDQ IDIV,ARPL,INS,JO	STC,STI ROR RET – RETF PUSHA – PUSHAD LOOP,JNC JC,CWD CLI,BT XCHG,NEG ADC POPA – POPAD SBB

۴- ارزیابی

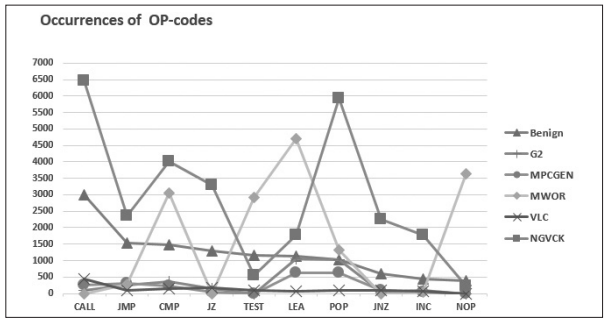
در روش پیشـــنهادی این مقاله توجـــه اصلی به بهبود دقت شناسایی نسبت به مقاله های پیشین، افزایش کارایی سیســـتم از جمله میزان حافظه مصرفی و ســـرعت در حد قابل قبولی نسبت به روش‌های پیشین، پوشش نقاط ضعف روش‌های قبلی مانند جایگزینی دستورات مشابه که موجب کاهش درصد شناســـایی می‌شده اســـت، خواهد بود. حال بـــا مدنظر قرار دادن نکات مطرح شـــده و فرضیات روش پیشنهادی آزمایش‌ها و ارزیابی‌ها انجام خواهد شد.

٤-١- آماده‌سازی مجموعه داده

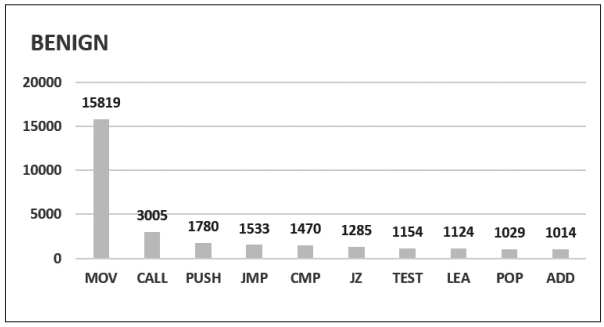
به‌منظور ارزیابی روش پیشـــنهادی، ٤٥٠ فایل متشکل از ٤٠ فایل ســـالم از مجموعه Cygwin و ٤١٠ فایل بدافزار فراریخت مورد بررسی قرار گرفته است. بدافزارها از پنج خانـــواده بدافزارهای فراریخت متشـــکل از ۲۵۰ ویروس NGVCK، ۵۰ ویـــروس G2، ۵۰ ویـــروس MPCGEN، ۱۰ ویروس VLC و ۵۰ کـــرم MWOR بوده‌اند. ابتدا همه ٤٥٠ فایل مورد بررســـی، با استفاده از ابزار IDA PRO[17] به کد اســـمبلی برگردانده می‌شوند. ســـپس کد هر یک از فایل‌ها به‌منظور شـــمارش ابتـــدا باید استانداردســـازی روی آن انجام شود. در این مرحله نویسه‌هایی که عملیات شمارش را با اختلال مواجه می‌ســـازند و در اصطلاح هیچ نقشـــی در ســـاختار کد ندارند از کد اصلی حذف می‌شوند و فقط ساختار اصلی کد باقی می‌ماند.

پایه و اســـاس روش پیشنهادی در شـــمارش کدهای عملیاتی اســـت که در ریزپردازنده ۸۰۸٦ در زبان اسمبلی ۱۹۱ کـــد عملیاتـــی وجـــود دارد. تمامی ۱۹۱ مـــورد برای تمامـــی ٤٥٠ فایل مورد بررســـی، در مرحله پیش‌پردازش شمارش می‌شـــوند. با آزمایش‌ها و ارزیابی‌های صورت گرفته و مشاهده تاثیر انتخاب حد آستانه در کارایی روش پیشـــنهادی، حد آســـتانه مقدار k=٥ تعیین شد که دلیل آن

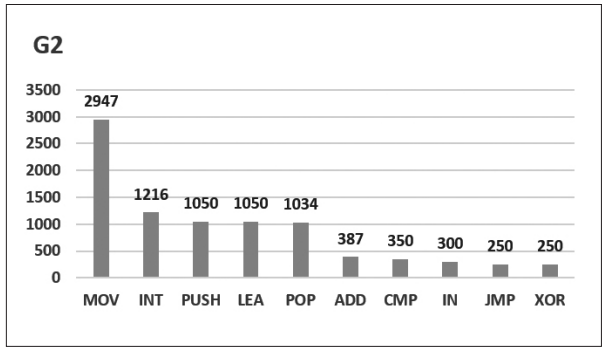
17-https://www.hex-rays.com/products/ida/support/download.shtml



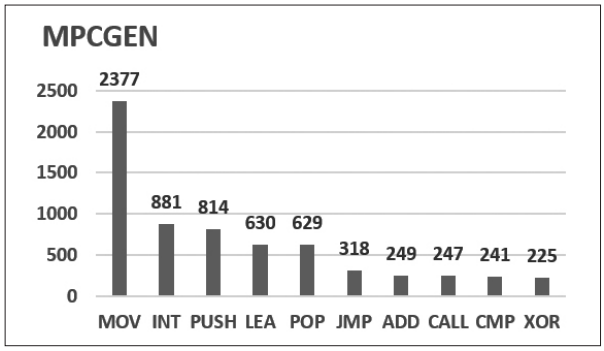
شکل۳: تفاوت در رخداد برخی از کدهای عملیاتی مختلف در خانواده‌های متفاوت بدافزارهای فراریخت و فایل‌های سالم



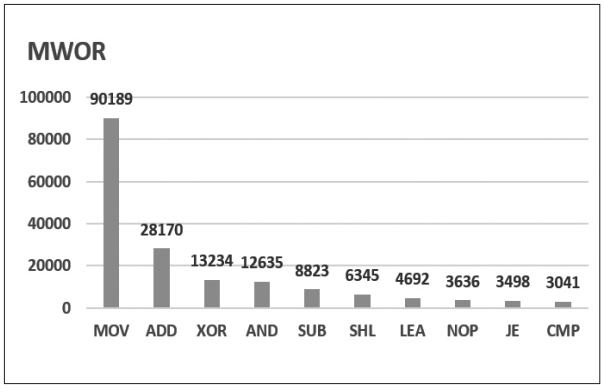
شکل۴: نرخ تکرار ۱۰ کد عملیاتی پر تکرار در فایل‌های سالم



شکل۵: نرخ تکرار ۱۰ کد عملیاتی پر تکرار در خانواده بدافزار G2



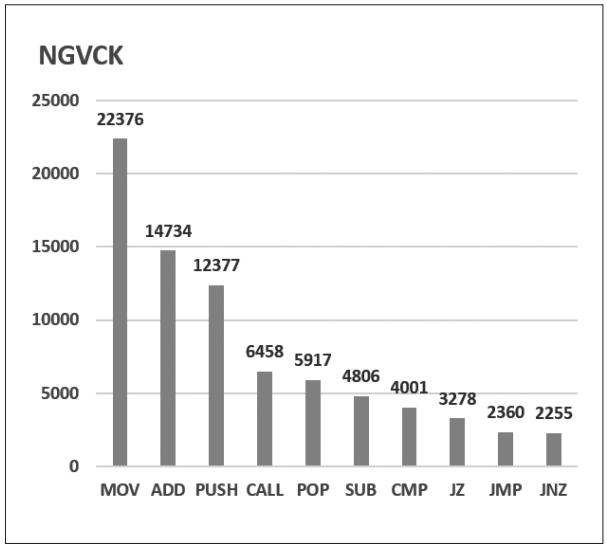
شکل۶: نرخ تکرار ۱۰ کد عملیاتی پر تکرار در خانواده بدافزار MPCGEN



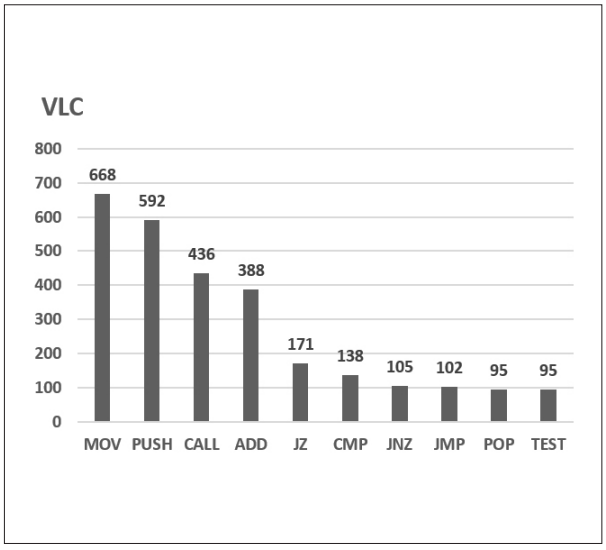
شکل۷: نرخ تکرار ۱۰ کد عملیاتی پر تکرار در خانواده بدافزار MWOR

این است که مقدار کمتر از پنج باعث حذف کدهای عملیاتی کمتری می‌شـــد که در واقع چون نرخ تکرار پایینی داشتند نقشی در روند شناسایی نداشتند و انتخاب مقداری بالاتر از پنج باعث افت دقت شناسایی شده بود. با در نظر گرفتن k=۵، تعداد کدهای عملیاتی که پالایش و انتخاب شدند n=۸۳ می‌باشـــد و تعداد کدهای عملیاتی حذف شده نیز m=۱۰۷ خواهد بود. با مشاهده نرخ تکرار کدهای عملیاتی و ثبات‌ها در فایل‌های سالم و خانواده‌های مختلف از بدافزارها، کاملا مشـــهود بود که تمایزی بین تعـــداد تکرارها برای کدهای عملیاتـــی مختلف وجود دارد. ایـــن تمایز در تعداد تکرارها پایه و اســـاس روش پیشنهادی این مقاله است. با توجه به شکل (۳) این تمایز به خوبی نشان داده شده است:

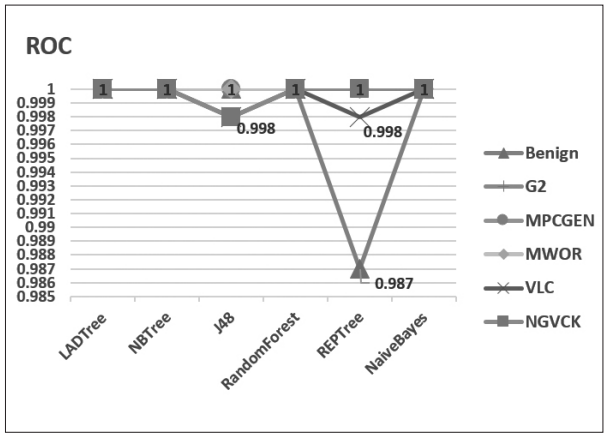
مقایسه تحلیل‌ها نشان می‌دهد که توزیع کدهای عملیاتی در برنامه‌های ســـالم با توزیع آن‌ها در بدافزارها متفاوت اســـت. همچنین میزان نرخ تکرار ۱۰ کـــد عملیاتی پرتکرار در فایل‌های ســـالم و پنج خانواده بدافزار فراریخت که در شکل‌های (۴،۵،۶،۷،۸،۹) نشان داده شده‌اند بیانگر این مسئله است که در هر یک از خانواده‌ها، کدهای عملیاتی متفاوتی با نرخ تکرار متفاوتی پرتکرار هستند و این می‌تواند تمایزی را برای طبقه‌بندی کـــردن خانواده‌های مختلف ایجاد کند. برخی از کدهای عملیاتی فقط در برنامه‌های ســـالم هستند و هرگز در بدافزارها وجود ندارند. بعضی کدهای عملیاتی ممکن اســـت در بدافزارها وجود داشته باشند اما هرگز در برنامه‌های ســـالم وجود ندارند. برخی از کدهای عملیاتی نه در بدافزارها وجود دارند و نه در برنامه‌های ســـالم که این موارد در ادامه معیارهایی برای طبقه‌بندی‌ها در روش پیشـــنهادی شـــده اســـت. به‌عنوان مثال برخی از کدهای



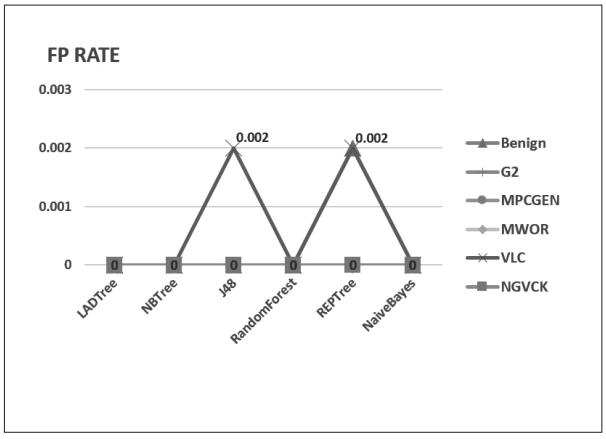
شکل۸: نرخ تکرار ۱۰ کد عملیاتی پر تکرار در خانواده بدافزار NGVCK



شکل۹: نرخ تکرار ۱۰ کد عملیاتی پر تکرار در خانواده بدافزار VLC



شکل۱۰: مقایسه میزان ROC برای پنج خانواده بدافزارهای فراریخت و فایل‌های سالم



شکل۱۱: مقایسه میزان نرخ مثبت کاذب برای پنج خانواده بدافزارهای فراریخت و فایل‌های سالم

عملیاتی مانند AND, POP, PUSH, ADD, MOV و مواردی که در شکل‌ها مشهود است دارای نرخ بالای رخداد در برنامه‌های سالم و همچنین دارای تنوع وسیعی هستند درحالی‌که کدهای عملیاتی دیگر مقادیر رخداد و تنوع آن‌ها اختلاف نزدیکی دارد.

٤-۲- تحلیل نتایج

با توجه به تحلیل‌های صورت گرفته و استفاده از الگوریتم‌های دسته‌بندی مختلف، شکل‌های (۱۰،۱۱،۱۲،۱۳) به دست آمده‌اند و نتایج کلی در جدول (۲) نشان داده شده است. تحلیل نتایج طبقه‌بندی‌ها در جدول (۲) نشان می‌دهد که روش پیشنهادی قادر است که به‌طور صحیحی طبقه‌بندی بین بدافزارهای فرایخت و فایل سالم را با دقت بسیار بالا انجام دهد. علاوه بر این نرخ مثبت کاذب برای برنامه سالم و بدافزارهای تولید شده با همه موتورهای تکثیر بسیار پایین و تقریبا صفر است. نتیجه شایان توجه این است که روش پیشنهادی در چهار الگوریتم دسته‌بندی دارای دقت شناسایی ۱۰۰ درصدی است و نرخ مثبت کاذب در این چهار الگوریتم مقدار صفر را نشان می‌دهد.

نکات حائز اهمیت در مورد نتایج به‌دست آمده این است که در بسیاری از حالات شناسایی ۱۰۰ درصدی را نشان

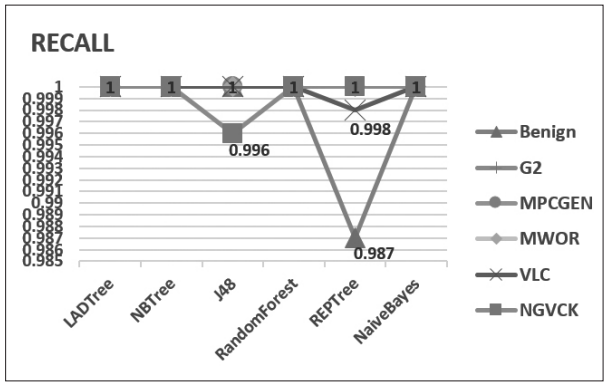
جدول ۲: نتایج مرحله طبقه‌بندی برای شش الگوریتم طبقه‌بندی

فایل‌ها	ناحیه ROC	معیار F	بازیابی	دقت	مثبت کاذب	مثبت واقعی	الگوریتم‌ها
BENIGN	۱	۱	۱	۱	۰	۱	LADTree
G2	۱	۱	۱	۱	۰	۱	
MPCGEN	۱	۱	۱	۱	۰	۱	
MWOR	۱	۱	۱	۱	۰	۱	
VLC	۱	۱	۱	۱	۰	۱	
NGVCK	۱	۱	۱	۱	۰	۱	
BENIGN	۱	۱	۱	۱	۰	۱	NBTree
G2	۱	۱	۱	۱	۰	۱	
MPCGEN	۱	۱	۱	۱	۰	۱	
MWOR	۱	۱	۱	۱	۰	۱	
VLC	۱	۱	۱	۱	۰	۱	
NGVCK	۱	۱	۱	۱	۰	۱	
BENIGN	۱	۱	۱	۱	۰	۱	J48
G2	۱	۱	۱	۱	۰	۱	
MPCGEN	۱	۱	۱	۱	۰	۱	
MWOR	۱	۱	۱	۱	۰	۱	
VLC	۰/۹۹۸	۰/۹۵۲	۱	۰/۹۰۹	۰/۰۰۲	۱	
NGVCK	۰/۹۹۸	۰/۹۹۸	۰/۹۰۶	۱	۰	۰/۹۹۶	
BENIGN	۱	۱	۱	۱	۰	۱	RandomForest
G2	۱	۱	۱	۱	۰	۱	
MPCGEN	۱	۱	۱	۱	۰	۱	
MWOR	۱	۱	۱	۱	۰	۱	
VLC	۱	۱	۱	۱	۰	۱	
NGVCK	۱	۱	۱	۱	۰	۱	
BENIGN	۰/۹۸۷	۰/۹۷۵	۰/۹۷۵	۰/۹۷۵	۰/۰۰۲	۰/۹۷۵	REPTree
G2	۱	۱	۱	۱	۰	۱	
MPCGEN	۱	۱	۱	۱	۰	۱	
MWOR	۱	۱	۱	۱	۰	۱	
VLC	۰/۹۹۸	۰/۹	۰/۹	۰/۹	۰/۰۰۲	۰/۹	
NGVCK	۱	۱	۱	۱	۰	۱	
BENIGN	۱	۱	۱	۱	۰	۱	NaiveBayes
G2	۱	۱	۱	۱	۰	۱	
MPCGEN	۱	۱	۱	۱	۰	۱	
MWOR	۱	۱	۱	۱	۰	۱	
VLC	۱	۱	۱	۱	۰	۱	
NGVCK	۱	۱	۱	۱	۰	۱	

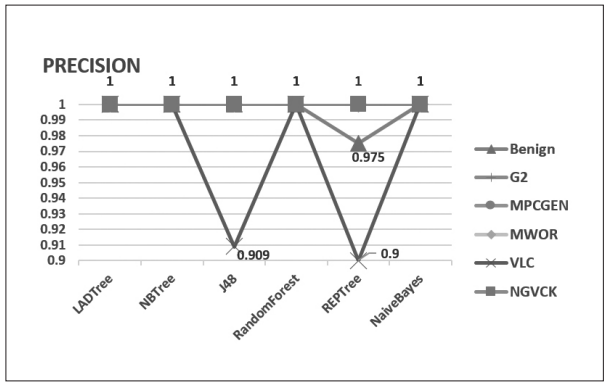
جدول ۳: مقایسه روش پیشنهادی این مقاله با روش پیشنهادی مقاله [۴]

بدافزارها	ناحیه ROC		نرخ مثبت کاذب		الگوریتم‌ها
	روش پیشنهادی	مقاله [۴]	روش پیشنهادی	مقاله [۴]	
BENIGN	۱	۰/۹۹	۰	۰/۰۱۲	LADTree
G2	۱	۰/۹۹	۰	۰/۰۰۵	
MPCGEN	۱	۰/۹۹	۰	۰/۰۰۵	
NGVCK	۱	۱	۰	۰/۰۰۸	
BENIGN	۱	۰/۹۹	۰	۰/۰۰۳	NBTree
G2	۱	۰/۹۹	۰	۰/۰۰۴	
MPCGEN	۱	۰/۹۹	۰	۰/۰۰۷	
NGVCK	۱	۱	۰	۰	
BENIGN	۱	۰/۹۹	۰	۰/۰۱۳	J48
G2	۱	۰/۹۹	۰	۰/۰۰۱	
MPCGEN	۱	۰/۹۹	۰	۰/۰۰۴	
NGVCK	۰/۹۹۸	۱	۰	۰	
BENIGN	۱	۰/۹۹	۰	۰	RandomForest
G2	۱	۰/۹۹	۰	۰/۰۰۳	
MPCGEN	۱	۱	۰	۰/۰۰۱	
NGVCK	۱	۱	۰	۰/۰۰۳	
BENIGN	۰/۹۸۷	۰/۹۹	۰/۰۰۲	۰/۰۰۵	REPTree
G2	۱	۰/۹۹	۰	۰/۰۰۵	
MPCGEN	۱	۰/۹۹	۰	۰/۰۰۴	
NGVCK	۱	۰/۹۹	۰	۰/۰۰۳	

می‌دهد که نشان‌دهنده این مسئله است که نفوذ بدافزارهای فراریخت تقریبا صفر خواهد بود. همچنین میزان نرخ مثبت کاذب در بسیاری از حالات برابر با صفر بوده است که باز هم بسیار مطلوب است. با توجه به نتایج به‌دست آمده، در چهار الگوریتم طبقه‌بندی LADTree, NBTree, Random-Forest, Naivebayes مقدار ROC=1 و FP=0 می‌باشد که نشان‌دهنده عملکرد بسیار مطلوب روش پیشنهادی است. در دو الگوریتم طبقه‌بندی دیگر نیز نتایج مطلوبی کسب شده است که جای بهبود نیز دارد. همچنین کارایی روش پیشنهادی که بسیار دارای اهمیت است نسبت به مقالات مشابه افزایش پیدا کرده است که با در نظر گرفتن میزان تعداد کدهای عملیاتی کمتر که در حدود نصف مقاله [٤] بوده است و با توجه به نصف شدن تعداد کدهای عملیاتی شمارش شده، سرعت شناسایی نیز افزایش پیدا کرده است که این بهبود نیز بسیار مطلوب است. در مقاله‌هایی که در بخش کارهای مرتبط نیز بحث شد یکی از نقاط ضعف این بود که با جایگزین کردن دستورات مشابه مانند JE و JZ و یا REPE و REPZ که دقیقا یک عملیات را انجام می‌دهند اما ساختار لغوی متفاوت دارند روش پیشنهادی آن‌ها دچار شکست می‌شده است اما برای روش پیشنهادی این مقاله، تمامی دستورات مشابه در شمارش‌ها یکسان در نظر گرفته شده‌اند و با جایگزین کردن آن‌ها هیچ تاثیری در



شکل۱۲: مقایسه میزان بازیابی برای پنج خانواده بدافزارهای فراریخت و فایل‌های سالم



شکل۱۳: مقایسه میزان دقت برای پنج خانواده بدافزارهای فراریخت و فایل‌های سالم

دقت شناسایی روش پیشنهادی مشاهده نمی‌شود که بهبود مناسبی نسبت به مقالات پیشین است.

نکته دیگر هم این‌که در مقالات مختلف در حوزه شناسایی بدافزارهای فراریخت معمولا یک معیار یا شاخص برای شناسایی انتخاب می‌شود که اگر نویسندگان بدافزارهای فراریخت بتوانند آن معیار را دچار نقص و چالش کنند روش پیشنهادی با شکست روبرو خواهد شد. اما در روش پیشنهادی این مقاله چهار معیار متفاوت برای شناسایی و طبقه‌بندی خانواده‌های مختلف در نظر گرفته شده‌اند که حتی با شکست یکی از معیارها باز هم تاثیر ناچیزی در روند شناسایی کلی انجام می‌گیرد و روش با شکست مواجه نمی‌شود که نکته‌ای حائز اهمیت است. در جدول (۳) بهبودهای حاصل شده نسبت به مقاله]٤[که نتایجی بسیار مطلوب را کسب کرده بود نشان داده شده است. جدول (۳) نشان می‌دهد که تقریبا در تمامی حالات روش پیشنهادی این مقاله دارای بهبود بوده است.

۵- نتیجه‌گیری و کارهای آینده

روش پیشنهادی این مقاله با شمارش و تحلیل ثبات‌ها و کدهای عملیاتی به شناسایی بدافزارهای فراریخت می‌پردازد و نشان می‌دهد که نرخ تکرار کدهای عملیاتی مختلف در خانواده‌های متفاوتی از بدافزارهای فراریخت و فایل‌های سالم، ناهمسان است که این مسئله می‌تواند وجه تمایزی را ایجاد کند. نتایج نشان می‌دهد که این روش یک معیار موثر برای نشان دادن تمایز بین خانواده‌های مختلف بدافزارها و فایل‌های سالم است. یافته‌های آزمایش‌ها نمایانگر این است که با وجود سادگی، روش پیشنهادی بسیار دقیق می‌باشد. در روش پیشنهادی سربار محاسباتی بسیار کم است، ایده اساسی روش به سادگی قابل فهم و در ضد بدافزارها به آسانی قابل پیاده‌سازی است، در مقابل روش‌های جانشینی دستورات مشابه و فنون مختلف مبهم سازی با توجه به شناسایی بر مبنای معیارهای مختلف، مقاوم می‌باشد و دارای کارایی مطلوبی از نظر حافظه مصرفی و سرعت شناسایی می‌باشد. همچنین در مقایسه با روش‌های مشابه موجود دارای بهبودهایی در ابعاد مختلف است.

برای کارهای آینده می‌توان تحلیل پویا را نیز به روش پیشنهادی اضافه کرد تا قدرت این روش افزایش یابد چون در مواردی بدافزارهای فراریخت از روش‌هایی از مبهم سازی استفاده می‌کنند که بدنه کد بدافزار فراریخت رمزنگاری خواهد شد که باعث می‌شود روش‌های تحلیل ایستا به‌طور کامل دچار شکست شوند که ایده پیشنهادی برای کار آینده این است که با اضافه کردن بخش تحلیل پویا و مشاهده کردن[۱۸] اجرای آن‌ها در محیط‌های امن[۱۹] و مجازی[۲۰] می‌توان این نقص را هم پوشش داد که در کارهای آینده مدنظر خواهد بود.

18-Monitoring
19-Sandbox
20-Virtual Machine

مراجع

1- Canfora, G., Mercaldo, F., Visaggio, C. A., & Di Notte, P. Metamorphic malware detection using code metrics. Information Security Journal: A Global Perspective, 23(3), 57-67, 2014.

2- Aycock, J. Computer Viruses and Malware (Advances in Information Security). Secaucus, 2006.

3- Chen, L., Li, T., Abdulhayoglu, M., & Ye, Y. Intelligent malware detection based on file relation graphs. In Semantic Computing (ICSC), 2015 IEEE International Conference on (pp. 85-92). IEEE, 2015.

4- Canfora, G., Iannaccone, A. N., &Visaggio, C. A. Static analysis for the detection of metamorphic computer viruses using repeated-instructions counting heuristics. Journal of Computer Virology and Hacking Techniques, 10(1), 11-27, 2014.

5- Szor, P. The art of computer virus research and defense. Pearson Education, 2005.

6- Mehra, V., Jain, V., &Uppal, D. DaCoMM: Detection and Classification of Metamorphic Malware. In Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on (pp. 668-673). IEEE, 2015.

7- Kapoor, A., & Dhavale, S. Control Flow Graph Based Multiclass Malware Detection Using Bi-normal Separation. Defence Science Journal, 66(2), 138-145, 2016.

8- Rad, B. B., &Masrom, M. Metamorphic virus variants classification using opcode frequency histogram. arXiv preprint arXiv:1104.3228. 2011.

9- Runwal, N., Low, R. M., & Stamp, M. Opcode graph similarity and metamorphic detection. Journal in Computer Virology, 8(1-2), 37-52, 2012.

10- Toderici, A. H., & Stamp, M. Chi-squared distance and metamorphic virus detection. Journal of Computer Virology and Hacking Techniques, 9(1), 1-14, 2013.

11- Khammas, B. M., Monemi, A., Ismail, I., Nor, S. M., & Marsono, M. N. Metamorphic Malware Detection Based on Support Vector Machine Classification of Malware Sub-Signatures. TELKOMNIKA (Telecommunication Computing Electronics and Control), 14(3), 2016.

12- Al Daoud, E., Jebril, I. H., & Zaqaibeh, B. Computer virus strategies and detection methods. Int. J. Open Problems Compt. Math, 1(2), 12-20, 2008.

13- Pandey, S. K., &Mehtre, B. M. Performance of malware detection tools: A comparison. In Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on (pp. 1811-1817). IEEE, 2014.