

---

## Presenting a Malware Detection System by Implementing Hardware Counters Based on the Multi-Layer Perceptron Neural Network (MLP) and the Dragonfly Optimization Algorithm

M. Karami, M. Mosleh\*

\*Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran

(Received: 14/12/2019, Accepted: 05/08/2020)

### ABSTRACT

*Today, one of the most important challenges of information security and communication networks is the increasing number of malwares and, consequently, finding suitable ways to protect systems. Timely acknowledgement of the malicious effects of malwares and devising plans and procedures to deal with them are the most important challenges for programmers and information security professionals. Intelligent malware detection systems are able to model malicious behavior well. Extracting appropriate features and using efficient classifiers can improve the performance of such systems. In this paper, a new approach to malware detection is proposed using the synergy of the features of hardware counters and the optimization of the multilayer perceptron neural network classifier. The proposed system is able to identify healthy files from malware by extracting features with high discrimination and using the neural network optimized by the dragonfly algorithm. In order to evaluate the proposed system, a data set is used that includes 168 healthy samples and 437 samples infected with malware. The results of the simulations show the higher performance of the proposed category compared to other categories, with the proposed system being able to detect the presence of malware-infected files with 86% accuracy.*

**Keywords:** Malware, Hardware Counters, Multi-Layer Perceptron Neural Network (MLP), Dragonfly Optimization Algorithm

---

\*Corresponding Author Email: mosleh@iaud.ac.ir

## علمی - پژوهشی

## ارائه یک سامانه تشخیص بدافزار رفتاری بر اساس عملکرد شمارنده‌های سخت‌افزاری مبتنی بر شبکه عصبی و الگوریتم بهینه‌سازی سنجاقک

محمد کرمی<sup>۱</sup>، محمد مصلح<sup>۲\*</sup>

۱- دانشجوی کارشناسی ارشد گروه مهندسی کامپیوتر، موسسه آموزش عالی کارون، اهواز، ایران

۲- دانشیار گروه مهندسی کامپیوتر، واحد دزفول، دانشگاه آزاد اسلامی، دزفول، ایران

(دریافت: ۱۳۹۸/۰۹/۲۳، پذیرش: ۱۳۹۹/۰۵/۱۵)

## چکیده

امروزه یکی از مهم‌ترین چالش‌های امنیت اطلاعات و شبکه‌های ارتباطی، افزایش روزافزون انواع بدافزارها و به دنبال آن یافتن راه‌های مناسب جهت حفاظت سامانه‌ها در مقابل آن‌ها است. شناخت به وقت و یافتن راه‌های مقابله با آثار مخرب بدافزارها از مهم‌ترین چالش‌های برنامه‌نویسان و متخصصین امنیت اطلاعات می‌باشد به طوری که در سال‌های اخیر استفاده از الگوریتم‌های داده‌کاوی و هوش مصنوعی به‌عنوان یکی از روش‌های نوظهور و امیدوارکننده برای مقابله با بدافزارها کاربرد بسیاری داشته است. سامانه‌های تشخیص بدافزار هوشمند قادر هستند با مدل‌سازی رفتار بدافزارها آن‌ها را به خوبی شناسایی نمایند. استخراج ویژگی‌های مناسب و به‌کارگیری دسته‌بند کارآمد می‌تواند کارایی چنین سامانه‌هایی را بهبود ببخشد. در این مقاله رویکردی جدید جهت تشخیص بدافزار با استفاده از هم‌افزایی ویژگی‌های شمارنده‌های سخت‌افزاری و دسته‌بند شبکه عصبی پرسپترون چندلایه بهینه پیشنهاد می‌شود. سامانه پیشنهادی با استخراج ویژگی‌هایی با قابلیت تفکیک‌پذیری بالا و نیز استفاده از شبکه عصبی بهینه‌شده به‌وسیله الگوریتم سنجاقک قادر است به خوبی فایل‌های سالم را از مخرب شناسایی نماید. به‌منظور ارزیابی سامانه پیشنهادی از یک مجموعه داده شامل ۱۶۸ نمونه سالم و ۴۳۷ نمونه آلوده به بدافزار استفاده می‌شود. نتایج حاصل از شبیه‌سازی‌ها کارایی بالاتر دسته‌بند پیشنهادی را در مقایسه با سایر دسته‌بندها نشان می‌دهد به طوری که سامانه پیشنهادی توانسته است با دقت ۸۶ درصد وجود فایل‌های آلوده به بدافزار را تشخیص دهد.

**کلیدواژه‌ها:** بدافزار، شمارنده‌های سخت‌افزاری، شبکه عصبی پرسپترون چند لایه (MLP)، الگوریتم بهینه‌سازی سنجاقک

## ۱- مقدمه

رشد روزافزون بدافزارها، شناخت به‌موقع و یافتن راه‌های مقابله با اثرات مخرب آن‌ها از مهم‌ترین چالش‌های پیش‌روی برنامه‌نویسان و متخصصین امنیت اطلاعات می‌باشد. بنابراین شرکت‌های ضد بدافزاری به‌طور جدی به دنبال راه‌های بهتری برای شناسایی بدافزارها هستند [۵-۶]. برای شناسایی بدافزار از تحلیل ایستا و پویا استفاده می‌شود. در تحلیل ایستا، بدافزار بودن یک فایل مستقیم از روی الگوها و یا مشخصه‌هایی که از کد آن استخراج می‌گردد، تشخیص داده می‌شود. از جمله ویژگی‌هایی که برای تشخیص بدافزارها در روش‌های ایستا به‌کار می‌رود می‌توان به توالی بایت‌های فایل، دستورالعمل‌های برنامه، منابع درون فایل و توابع وارد شده و بیرون شده اشاره کرد. در تحلیل پویا، برای تشخیص بدافزار بودن یک فایل، بر اساس ویژگی‌های رفتاری برنامه در زمان اجرا مشخص می‌شود [۷]. از جمله ویژگی‌های رفتاری برنامه می‌توان به میزان مصرف سامانه، ارتباطات شبکه، فایل‌های خوانده شده و یا نوشته‌شده اشاره نمود. از مشکلات اساسی این روش این است که ممکن است بسیاری از

در سال‌های اخیر گسترش بدافزارها به‌صورت شگفت‌انگیزی افزایش یافته است. با توجه به رشد فزاینده‌ای که بدافزارها در سال‌های اخیر پیدا کرده‌اند می‌توان گفت رفتار بدافزارهای جدید در حال مبهم‌تر شدن و پیچیده‌تر شدن می‌باشد [۱-۲]. بدافزار کوتاه‌شده عبارت نرم‌افزارهای مخرب بوده و به معنای نرم‌افزارهایی است که می‌توانند برای سرقت اطلاعات، به خطر انداختن فعالیت‌های رایانه و دور زدن کنترل‌های دسترسی مورد استفاده قرار گیرند [۳]. بدافزارها بر اساس نوع حملاتشان به سامانه‌های رایانه‌ای انواع مختلفی از جمله ویروس<sup>۱</sup>، کرم<sup>۲</sup>، تروجان<sup>۳</sup>، باج‌افزار<sup>۴</sup>، روت‌کیت<sup>۵</sup> و غیره می‌باشند [۴]. با توجه به

\* رایانامه نویسنده مسئول: Mosleh@iaud.ac.ir

1 Virus

2 Worm

3 Trojan

4 Ransom ware

5 Root kit

داده‌ای که شامل ۲۷۰۰۰ بدافزار و ۸۳۸ نرم‌افزار بی‌خطر بود استفاده کردند. آن‌ها همچنین جهت تشخیص و دسته‌بندی الگوریتم جنگل تصادفی را به‌کار بردند. کاماس و همکاران [۱۶] در سال ۲۰۱۶ یک سامانه تشخیص بدافزار مبتنی بر تحلیل ایستای دودویی‌های بدافزار متامورفیک، بدون جداسازی پیشنهاد دادند. ویژگی‌ها از دودویی‌ها که می‌توانند به فرم بسته‌های بار در سامانه تشخیص شبکه یا فایل‌هایی در میزبان باشند، بر اساس  $n$ -gram استخراج می‌شدند. به‌منظور دسته‌بندی نیز از ماشین بردار پشتیبان استفاده شده است. گروسی و همکاران [۱۷] در سال ۲۰۱۶ از روش یادگیری عمیق جهت تشخیص بدافزار استفاده نمودند. آن‌ها به‌منظور کاهش ابعاد داده از رویکرد اطلاعات متقابل بهره گرفتند. در سال ۲۰۱۷ پاتل و همکاران [۱۸] به تحلیل کاملی از روش‌های یادگیری ماشین جهت دسته‌بندی برنامه‌های سالم و مخرب پرداختند. آن‌ها به‌منظور پیاده‌سازی نرم‌افزاری، تمامی دسته‌بندها را در هسته سامانه عامل به‌طور کامل پیاده‌سازی کردند تا تأثیرات نرم‌افزاری مختلف مشخص شود. علاوه بر این به‌منظور اجرای سریع، این الگوریتم‌ها را بر روی FPGA پیاده‌سازی نمودند. در سال ۲۰۱۷ ساینگ و همکاران [۱۹] یک تحلیل جامع از به‌کارگیری رویکردهای یادگیری ماشین و شمارنده‌های عملکرد سخت‌افزار را برای یک زیرمجموعه خاص از نرم‌افزارهای مخرب روتکیت‌های هسته ارائه دادند. علاوه‌براین آن‌ها از روش‌های انتخاب ویژگی جهت تعیین ویژگی‌های کارآمد استفاده نمودند. در سال ۲۰۱۸ صیادی و همکاران [۲۰] از روش‌های یادگیری جمعی جهت بهبود عملکرد سامانه‌های تشخیص بدافزار مبتنی بر عملکرد شمارنده‌های سخت‌افزاری استفاده کردند. برای این منظور آن‌ها از هشت مدل یادگیری ماشین مقاوم و دو دسته‌بند جمعی مشهور استفاده کردند. در سال ۲۰۱۸ رحمان و همکاران [۲۱] یک چارچوب ترکیبی کارآمد برای تشخیص بدافزار در برنامه‌های کاربردی اندروید ارائه کردند که از هر دو تحلیل اکتشافی و امضا بهره می‌گرفت. به‌منظور دسته‌بندی، از ماشین بردار پشتیبان، درخت تصمیم J48 و KNN استفاده کردند. نتایج به‌دست‌آمده نشان داد که ماشین بردار پشتیبان در حالت دودویی‌ها و KNN در حالت فایل‌های مینی‌فیست بهترین گزینه برای تشخیص بدافزار در دستگاه‌های اندروید هستند.

در این مقاله یک سامانه تشخیص بدافزار هوشمند پیشنهاد می‌شود که قادر است با یادگیری عملکرد ویژگی‌های شمارنده‌های سخت‌افزاری توسط شبکه عصبی بهینه‌شده به‌وسیله الگوریتم سنجاکف فایل‌های سالم را از بدافزار تشخیص دهد. جهت یافتن اوزان و بایاس‌های مناسب شبکه عصبی پرسپترون چندلایه از الگوریتم فرامکاشفه‌ای سنجاکف استفاده شده است.

بدافزارها، در هر محیط اجرایی، رفتار بدخواهانه خود را نشان ندهند [۸].

امروزه یکی از مهم‌ترین چالش‌های امنیت اطلاعات و شبکه‌های ارتباطی، افزایش روزافزون بدافزارها و به دنبال آن یافتن راه‌های مناسب جهت حفاظت سامانه‌ها در مقابل آنهاست که شناخت به‌موقع آن‌ها و یافتن راه‌های مقابله با اثرات مخرب بدافزارها از مهم‌ترین دغدغه‌های برنامه‌نویسان و متخصصین امنیت اطلاعات می‌باشد. بنابراین در طول سال‌های گذشته، روش‌های متنوعی به‌منظور شناسایی فعالیت‌های بدافزارها ارائه شده است که در ادامه به اختصار به چند مورد اشاره می‌شود.

در سال ۲۰۱۴، تانگ و همکاران [۹] یک سامانه تشخیص بدافزار مبتنی بر امضا با استفاده از ویژگی‌های سخت‌افزاری پیشنهاد کردند. آن‌ها نشان دادند که سامانه پیشنهاد شده قادر است طیف وسیعی از بدافزارها را به‌درستی تشخیص دهد. در سال ۲۰۱۵، کاساوانه و همکاران [۱۰] یک سامانه تشخیص بدافزار مبتنی بر الگوریتم‌های یادگیری ماشین با استفاده از ویژگی‌های سخت‌افزاری سطح پایین پیشنهاد کردند. آن‌ها از یک دسته‌بند جمعی استفاده کردند که قادر بود نرخ تشخیص بدافزار را در مقایسه با دسته‌بندهای منفرد بهبود ببخشد. در سال ۲۰۱۶، فان و همکاران [۱۱] یک چارچوب کارآمد مبتنی بر روش کاوش دنباله جهت تشخیص بدافزار پیشنهاد کردند. الگوریتم کاوش الگوی ترتیبی قادر بود الگوهای تفکیک‌پذیری از برنامه‌های سالم و مخرب استخراج نماید. سپس از یک دسته‌بند  $k$ -نزدیکترین همسایه جدید به‌عنوان دسته‌بند استفاده کردند. در سال ۲۰۱۶، اوزسوی و همکاران [۱۲] پردازشگرهای سخت‌افزاری (MAP) تقویت‌شده با یک تشخیص‌دهنده مبتنی بر سخت‌افزار برخط را به‌عنوان اولین مرحله دفاع جهت تشخیص بدافزار از برنامه سالم به‌کار گرفتند. این سامانه قادر بود در خروجی نحوه به‌کارگیری رویکردهای مبتنی بر نرم‌افزار را اولویت‌بندی کند. در سال ۲۰۱۶، هدا و همکاران [۱۳] یک رویکرد تشخیص بدافزار مستقل از امضا پیشنهاد کردند. رویکرد پیشنهادی از یک روش ترکیبی مبتنی بر فیلتر حداکثر ارتباط-حداقل افزونگی و پوششی ماشین بردار پشتیبان و جهت انتخاب ویژگی‌ها بهره می‌گرفت. ایمران و همکاران [۱۴] در سال ۲۰۱۶ برای تشخیص بدافزارها از روش تحلیل نمادین استفاده کردند. روش پیشنهاد شده آن‌ها، روشی ترکیبی بود که از فراخوانی‌های سامانه استخراج شده از تحلیل ایستا و پویا به‌عنوان نماد استفاده می‌کرد. همچنین آن‌ها جهت دسته‌بندی، مدل مارکوف مخفی را به‌کار بردند. هنسن و همکاران [۱۵] در سال ۲۰۱۶ برای تحلیل حجم انبوهی از بدافزارهایی که برای تحلیل ایستا مبهم بودند از سفارشی‌سازی جعبه شن KUCKOO با استفاده از مجموعه

بایاس نورون هستند. پارامترهای  $w$  و  $b$  قابل تنظیم می‌باشند. در حقیقت یادگیری به این معنی است که  $w$  و  $b$  طوری تغییر کنند که رابطه ورودی و خروجی نورون با هدف خاصی مطابقت کند.

## ۲-۲- الگوریتم بهینه‌سازی سنجاک

الگوریتم بهینه‌سازی سنجاک برای اولین بار توسط میرجلیلی در سال ۲۰۱۶ مطرح شد [۲۴]. ایده اصلی الگوریتم بهینه‌سازی سنجاک از رفتارهای هوش جمعی ایستا<sup>۲</sup> و هوش جمعی پویا<sup>۳</sup> سرچشمه می‌گیرد. این دو رفتار هوش جمعی مشابه دو مرحله اصلی از بهینه‌سازی با الگوریتم‌های فرااکتشافی یعنی اکتشاف و استخراج می‌باشند. سنجاک‌ها با تشکیل گروه‌های کوچک‌تر در نواحی مختلف به صورت یک گروه ایستا پرواز می‌کنند که این کار هدف اصلی در مرحله اکتشاف است. در گروه پویا، سنجاک‌ها در گروه‌های بزرگتر و در امتداد یک جهت پرواز می‌کنند که یک رفتار مطلوب در مرحله استخراج می‌باشد. با توجه به این دو رفتار می‌توان نتیجه گرفت که پنج عامل اصلی در بهروزرسانی موقعیت نمونه در رفتار ازدحامی وجود دارد. این پنج عامل شامل تفکیک یا مجزا بودن، هم‌ترازی یا حرکت اصولی، انسجام یا پیوستگی، جذب به سمت شکار و دوری از دشمن هستند که مدل ریاضی هر یک از این رفتارها به صورت زیر بیان می‌شود:

### ۱) مرحله تفکیک

مدل سازی ریاضی مرحله تفکیک با استفاده از رابطه (۲) انجام می‌شود که در آن  $X$  نشان‌دهنده موقعیت فعلی سنجاک،  $X_j$  نشان‌دهنده زامین همسایه سنجاک و  $N$  تعداد سنجاک‌های همسایه را نشان می‌دهد.

$$S_i = - \sum_{j=1}^N (X - X_j) \quad (2)$$

### ۲) مرحله هم‌ترازی

مدل سازی ریاضی مرحله هم‌ترازی با استفاده از رابطه (۳) انجام می‌شود که در آن  $V_j$  نشان‌دهنده سرعت زامین همسایه سنجاک است.

$$A_i = (\sum_{j=1}^N V_j) / N \quad (3)$$

### ۳) انسجام

مدل سازی ریاضی مرحله تفکیک با استفاده از رابطه (۴) انجام می‌شود که در آن  $X$  نشان‌دهنده موقعیت فعلی سنجاک،  $X_j$  نشان‌دهنده موقعیت زامین همسایه سنجاک و  $N$  برابر تعداد سنجاک‌های همسایه می‌باشد.

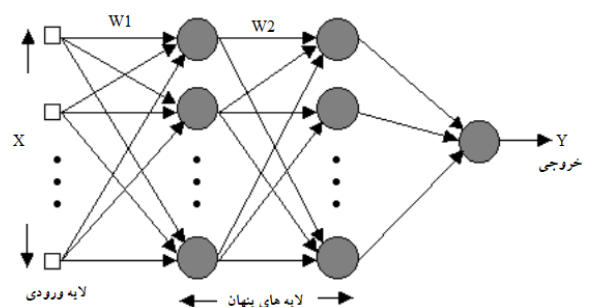
بخش‌بندی مقاله به شرح زیر است: در بخش دوم به معرفی شبکه عصبی پرسپترون چندلایه و الگوریتم بهینه‌سازی سنجاک پرداخته می‌شود. در بخش سوم سامانه تشخیص بدافزار پیشنهادی معرفی می‌شود. در بخش چهارم نتایج شبیه‌سازی‌ها و مقایسه‌ها ارائه می‌شود و در انتهای مقاله نیز به نتیجه‌گیری خواهیم پرداخت.

## ۲- معرفی ابزارهای پژوهش

در این بخش به معرفی دو مفهوم به کار گرفته شده در این مقاله شامل شبکه‌های عصبی پرسپترون چندلایه و نیز الگوریتم بهینه‌سازی سنجاک پرداخته می‌شود.

### ۲-۱- شبکه عصبی پرسپترون چند لایه

شبکه عصبی پرسپترون چندلایه (MLP)، دسته‌ای از شبکه‌های عصبی مصنوعی پیشخور می‌باشد که حداقل شامل سه لایه از گره‌ها است: یک لایه ورودی، یک لایه پنهان و یک لایه خروجی. به جز گره‌های ورودی، هر گره یک نورون است که از یک تابع فعال‌سازی غیرخطی استفاده می‌کند. شبکه عصبی MLP از روش یادگیری نظارت شده که پس انتشار نامیده می‌شود، برای آموزش استفاده می‌کند. شبکه عصبی MLP به وسیله لایه‌های متعدد و فعال‌سازی غیرخطی، از یک پرسپترون خطی متمایز شده و داده‌هایی که به صورت خطی قابل تفکیک نیستند متمایز می‌شوند. شکل (۱) ساختار یک شبکه عصبی MLP با بردار ورودی  $X$  و خروجی  $Y$  را نشان می‌دهد [۲۲-۲۳].



شکل (۱): ساختار یک شبکه عصبی MLP با دو لایه پنهان.

خروجی یک شبکه عصبی پرسپترون با بردار ورودی  $X$  و وزن‌های  $W$  و خروجی  $Y$  با استفاده از رابطه (۱) به دست می‌آید:

$$y = f(wx + b) \quad (1)$$

که در آن،  $f$  تابع محرک<sup>۱</sup> (فعالیت)،  $w$  و  $b$  به ترتیب وزن و

<sup>۲</sup> Static dragonfly swarms  
<sup>۳</sup> Dynamic dragonfly swarms

<sup>۱</sup> Activity function

۱۶۸ نمونه سالم تشکیل شده است. نمونه‌های بدافزار از سایت‌های virusign.com و tekdefense.com و نمونه‌های سالم از برنامه‌های کاربردی ویندوز جمع‌آوری شده‌اند. پس از اجرای هر یک از آن‌ها بر روی یک سامانه مجازی و ثبت تأثیرات آن‌ها بر روی رفتار شمارنده‌های سخت‌افزاری، یک مجموعه داده شامل ۶۰۶ الگو، هر یک با ۶۶ ویژگی و یک برجسب، استخراج می‌شود که به‌عنوان الگوهای رفتاری شمارنده‌های سخت‌افزاری در این مقاله مورد استفاده قرار می‌گیرند.

### ۳-۲-۳-۲-۳ نرمال‌سازی داده‌ها

نرمال‌سازی روشی برای قرار دادن داده‌های خارج از یک دامنه در یک دامنه مشابه است. بنابراین برای نرمال کردن داده‌ها در این مقاله از رابطه (۸) که هر یک از ویژگی‌ها را در بازه  $[-1, 1]$  قرار می‌دهد، استفاده می‌شود [۲۵]:

$$x_{normal} = 2 \frac{x - \min(X)}{\max(X) - \min(X)} - 1 \quad (8)$$

که  $x$  مقدار داده موردنظر جهت نرمال شدن،  $\min(x)$  کمینه بردار ورودی  $X$  و  $\max(X)$  بیشینه بردار ورودی  $X$  بوده و  $x_{normal}$  مقدار نرمال شده  $X$  است.

### ۳-۳-۳-۳ آموزش شبکه عصبی MLP

شبکه عصبی موردنظر در این مقاله از نوع MLP با یک لایه ورودی شامل ۶۶ نورون، یک لایه پنهان شامل ۲۰ نورون و یک لایه خروجی با یک نورون است. آموزش شبکه عصبی MLP با استفاده از الگوریتم پس انتشار انجام شده است به‌طوری‌که در آن متوسط مربعات خطای شبکه که با رابطه (۹) نشان داده شده است، کمینه می‌شود [۲۳]:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (9)$$

در رابطه (۹)،  $y_i$  شماره دسته واقعی نمونه نام و  $\hat{y}_i$  شماره دسته پیش‌بینی شده نمونه نام توسط شبکه عصبی MLP است. علاوه‌براین،  $n$  تعداد نمونه‌های آموزشی است. مقدار متوسط مربعات خطا در شبکه به انتخاب مقادیر وزن‌ها و بایاس‌های به‌کار گرفته شده بستگی دارد به‌طوری‌که انتخاب بهینه این مقادیر باعث کاهش متوسط خطای دسته‌بندی می‌شود.

در ادامه قصد داریم مقادیر بهینه وزن‌ها و بایاس‌ها را جهت رسیدن به بهترین کارایی، با استفاده از الگوریتم بهینه‌سازی سنجاکف به‌دست آوریم.

$$C_i = \frac{\sum_{j=1}^N X_j}{N} - X \quad (4)$$

### ۴ جذب به سمت شکار

مدل‌سازی ریاضی مرحله جذب به سمت شکار با استفاده از رابطه (۵) بیان می‌شود که در آن  $X$  نشان‌دهنده موقعیت فعلی سنجاکف و  $X^+$  نشان‌دهنده موقعیت غذا است.

$$F_i = X^+ - X \quad (5)$$

### ۵ دوری از دشمن

مدل‌سازی ریاضی مرحله دوری از دشمن با استفاده از رابطه (۶) انجام می‌شود که در آن  $X$  نشان‌دهنده موقعیت فعلی سنجاکف و  $X^-$  موقعیت دشمن را نشان می‌دهد.

$$E_i = X^- + X \quad (6)$$

رفتار سنجاکف‌ها ترکیبی از این پنج گروه است. به‌منظور به‌روزرسانی موقعیت سنجاکف‌ها در فضای جستجو و شبیه‌سازی حرکت‌های آن‌ها، بردار گام به‌صورت رابطه (۷) به‌روزرسانی می‌شود:

$$X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + wX_t \quad (7)$$

که در آن،  $s$  نشان‌دهنده ضریب تفکیک و  $S_i$  نشان‌دهنده مجزا بودن آامین سنجاکف،  $a$  نشان‌دهنده ضریب هم‌ترازی،  $A_i$  نشان‌دهنده حرکت اصولی آامین سنجاکف،  $c$  نشان‌دهنده ضریب انسجام،  $C_i$  نشان‌دهنده انسجام آامین سنجاکف،  $f$  ضریب جذب منبع غذا،  $F_i$  نشان‌دهنده منبع غذای آامین سنجاکف،  $e$  نشان‌دهنده ضریب دوری از دشمن،  $E_i$  نشان‌دهنده موقعیت دشمن از آامین سنجاکف و  $w$  نشان‌دهنده ضریب اینرسی است. در این رابطه  $X_t$  و  $X_{t+1}$  مکان سنجاکف‌ها در تکرارهای  $t$  و  $t+1$  هستند.

### ۳- ارائه روش پیشنهادی

در این بخش به معرفی و تشریح رویکرد پیشنهادی به‌منظور تشخیص خودکار بدافزار پرداخته خواهد شد. برای این منظور ابتدا مجموعه داده مورد استفاده معرفی می‌شود و سپس به تشریح شبکه عصبی مورد استفاده و نحوه بهینه‌سازی آن توسط الگوریتم بهینه‌سازی سنجاکف پرداخته می‌شود.

### ۳-۱- معرفی مجموعه داده

مجموعه داده مورد استفاده در این مقاله از ۴۳۷ نمونه بدافزار و

		دسته واقعی	
		سالم	بدافزار
دسته پیش بینی شده	سالم	TN	FN
	بدافزار	FP	TP

شکل (۲): ماتریس درهم ریختگی.

$TP^6$  : عبارت است از تعداد نمونه‌هایی که توسط روش پیشنهادی به درستی بدافزار تشخیص داده شده‌اند.

$FN^7$  : عبارت است از تعداد نمونه‌هایی که توسط روش پیشنهادی به اشتباه سالم تشخیص داده شده‌اند.

$FP^8$  : عبارت است از تعداد نمونه‌هایی که توسط روش پیشنهادی به اشتباه بدافزار تشخیص داده شده‌اند.

$TN^9$  : عبارت است از تعداد نمونه‌هایی که توسط روش پیشنهادی به درستی سالم تشخیص داده شده‌اند.

در ادامه با توجه به پارامترهای معرفی شده، معیارهای مورد نظر به صورت زیر معرفی می‌شوند:

معیار دقت نسبت تعداد نمونه‌های درست تشخیص داده شده توسط دسته‌بند به کل تعداد نمونه‌هاست و طبق رابطه (۱۱) بیان می‌شود.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

معیار حساسیت نسبت تعداد نمونه‌های مثبت درست تشخیص داده شده توسط دسته‌بند به تعداد کل نمونه‌های مثبت می‌باشد و طبق رابطه (۱۲) محاسبه می‌گردد.

$$Sensitivity = \frac{TP}{TP + FN} \quad (12)$$

معیار ویژگی نسبت تعداد نمونه‌های منفی درست تشخیص داده شده توسط دسته‌بند به تعداد کل نمونه‌های منفی می‌باشد و طبق رابطه (۱۳) بیان می‌شود.

$$Specificity = \frac{TN}{TN + FP} \quad (13)$$

معیار صحت نسبت تعداد نمونه‌های مثبت درست تشخیص داده شده توسط دسته‌بند به تعداد کل نمونه‌هایی است که به درستی یا به طور نادرست، مثبت تشخیص داده شده‌اند و توسط

### ۳-۴- بهینه‌سازی شبکه عصبی MLP با استفاده از الگوریتم سنجاک

اولین گام از بهینه‌سازی شبکه عصبی MLP با استفاده از الگوریتم سنجاک، کدگذاری مسئله است. بدین منظور موقعیت هر سنجاک با مقادیر وزن‌ها و بایاس با استفاده از رابطه (۱۰) تعریف می‌شود:

$$x = [w_1 \cdot w_2 \dots w_n, b_1, b_2 \dots b_m] \quad (10)$$

که  $w_i$  و  $b_i$  به ترتیب به وزن‌ها و بایاس‌های شبکه عصبی اشاره دارند.

در مرحله دوم جمعیت اولیه‌ای شامل ۵۰ سنجاک به صورت تصادفی در بازه  $[-1, 1]$  مقداردهی می‌شود.

در مرحله سوم برای هر عضو جمعیت، یک شبکه عصبی MLP آموزش می‌بیند و سپس مقدار متوسط مربعات خطای آن با استفاده از رابطه (۹) محاسبه می‌شود. مقدار MSE به عنوان هزینه هر راه حل منظور می‌شود. به عبارت دیگر راه‌حلی دارای میزان شایستگی بیشتری است که دارای مقدار MSE کمتری باشد.

در مرحله چهارم بهترین و بدترین راه‌حل‌ها به ترتیب به عنوان "منبع غذا" و "دشمن" در نظر گرفته می‌شوند.

در مرحله پنجم به منظور پوشش محلی و سراسری، موقعیت سنجاک‌ها تحت تأثیر عملیات تفکیک، هم‌ترازی، انسجام، جذب به سمت شکار و دوری از دشمن قرار گرفته و به روزرسانی می‌شوند (بر اساس معادلات (۲) تا (۷)).

در مرحله ششم مجدداً MSE راه‌حل‌های جدید محاسبه شده و "منبع غذا" و "دشمن" به روزرسانی می‌شوند.

مراحل پنجم و ششم به تعداد ۱۰۰۰ بار تکرار می‌شوند و در نهایت بهترین سنجاک به عنوان بهترین وزن‌ها و بایاس‌های شبکه MLP نتیجه می‌شود.

### ۴- ارزیابی روش پیشنهادی

در این بخش به ارزیابی روش پیشنهادی به منظور تشخیص فایل‌های آلوده به بدافزار پرداخته می‌شود. بدین منظور در ابتدا به معرفی ماتریس درهم ریختگی<sup>۱</sup> پرداخته و سپس معیارهای دقت<sup>۲</sup>، حساسیت<sup>۳</sup>، ویژگی<sup>۴</sup>، صحت<sup>۵</sup>، F1 معرفی می‌شوند [۲۶]. ماتریس درهم ریختگی به صورت شکل (۳) است:

<sup>6</sup> True Positive  
<sup>7</sup> False Negative  
<sup>8</sup> False Positive  
<sup>9</sup> True Negative

<sup>1</sup> Confusion matrix  
<sup>2</sup> Accuracy  
<sup>3</sup> Sensitivity  
<sup>4</sup> Specificity  
<sup>5</sup> Precision

رابطه (۱۴) محاسبه می‌شود.

$$F1 = 2 \times \left( \frac{\text{sensitivity} \times \text{precision}}{\text{sensitivity} + \text{precision}} \right) \quad (15)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (14)$$

لازم به ذکر است که از اعتبارسنجی متقابل برای محاسبه معیارهای ارزیابی روش پیشنهادی استفاده گردیده است که میانگین نتایج آن‌ها برای ۲۰ بار اجرای الگوریتم در جدول (۱) نشان داده شده است.

F1 معیاری است که میانگین هارمونی وزن‌دار دو معیار صحت و حساسیت را بیان می‌کند و طبق رابطه (۱۵) تعریف می‌شود.

جدول (۱): میانگین نتایج ارزیابی روش پیشنهادی در ۲۰ بار اجرای برنامه.

اعتبار سنجی متقابل	صحت	ویژگی	حساسیت	دقت	F1
K=2	۰/۹۱۶۷	۰/۹۹۵۲	۰/۱۶۴۲	۰/۷۹۴۲	۰/۲۷۸۵
K=5	۰/۸۷۵۰	۰/۹۸۸۱	۰/۲۵۹۳	۰/۸۱۰۸	۰/۴۰۰
K=7	۰/۹۰۰	۰/۹۸۳۳	۰/۴۷۳۷	۰/۸۶۰۸	۰/۶۲۰۷
K=10	۱	۱	۰/۳۰۷۷	۰/۸۳۶۴	۰/۴۷۰۶

در ادامه با استفاده از مجموعه داده مورد استفاده در این تحقیق، به مقایسه روش پیشنهادی با دیگر الگوریتم‌های دسته‌بندی پرداخته شد که در تمامی دسته‌بندی‌های مورد مقایسه نحوه تقسیم‌بندی داده‌ها بر اساس K=7 می‌باشد. نتایج به دست آمده بر اساس معیارهای مورد بررسی برای داده‌های آزمایشی، در جدول (۲) آمده است.

همان‌گونه که در جدول (۱) نشان داده شده است روش پیشنهادی با اعتبارسنجی‌های K=2, K=5, K=7, K=10 اجرا گردید که مقدار دقت تشخیص روش پیشنهادی جهت تشخیص بدافزار در اعتبارسنجی K=7 نسبت به سایر اعتبارسنجی‌ها بیشتر می‌باشد.

جدول (۲): مقایسه نتایج روش پیشنهادی با سایر دسته‌بندها.

دسته‌بند	صحت	ویژگی	حساسیت	دقت	F1 معیار
ماشین بردار پشتیبان با هسته RBF	۰/۵۳۹۴	۰/۹۱۶۲	۰/۳۰۳۷	۰/۷۶۶۷	۰/۳۸۸۶
ماشین بردار پشتیبان با هسته خطی	۱	۱	۰/۰۲۹۶	۰/۷۶۳۱	۰/۰۵۷۵
k نزدیکترین همسایه	۰/۴۲۷۲	۰/۶۹۸۵	۰/۶۹۶۳	۰/۶۹۸۰	۰/۵۲۹۵
درخت تصمیم	۰/۴۳۲۴	۰/۷۹۹۰	۰/۴۷۷۴	۰/۷۱۹۷	۰/۴۵۲۳
شبکه عصبی آبخاری	۰/۸۱۸۱	۰/۹۹۰۴	۰/۱۳۳۳	۰/۷۸۱۱	۰/۲۲۹۳
رگرسیون لجستیک	۰/۲۴۴۱	۰	۱	۰/۲۴۴۱	۰/۳۹۲۴
شبکه عصبی MLP	۰/۷۶۶	۰/۹۸۳۵	۰/۱۷۰۳	۰/۷۸۴۸	۰/۲۷۸۷
جنگل تصادفی	۰/۸۳	۰/۹۰۲۱	۰/۷۶۸۵	۰/۷۹۰	۰/۳۹۹۰
روش پیشنهادی	۰/۹	۰/۹۸۳۳	۰/۴۷۳۷	۰/۸۶۰۸	۰/۶۲۰۷

به میزان ۱۰/۲ درصد، رگرسیون لجستیک<sup>۶</sup> به میزان ۲۵/۲ درصد، شبکه عصبی<sup>۷</sup> MLP به میزان ۹/۶۸ درصد و در نهایت جنگل تصادفی<sup>۸</sup> به میزان ۸/۹۶ درصد بهبود را نشان می‌دهد. همچنین بالاتر بودن مقدار معیار F1 در دسته‌بندی پیشنهادی نسبت به سایر دسته‌بندی‌های مورد بررسی نشان از عملکرد مطلوب روش پیشنهادی دارد.

نتایج بررسی‌ها نشان می‌دهد که روش پیشنهادی نسبت به دسته‌بندی‌های ماشین بردار پشتیبان با هسته<sup>۱</sup> RBF به میزان ۱۲/۲۷ درصد، ماشین بردار پشتیبان با هسته خطی<sup>۲</sup> به میزان ۸/۱۲ درصد، k- نزدیکترین همسایه<sup>۳</sup> به میزان ۲۳/۳۲ درصد، درخت تصمیم<sup>۴</sup> به میزان ۱۹/۹ درصد، شبکه عصبی آبخاری<sup>۵</sup>

<sup>۶</sup> Logistic Regression

<sup>۷</sup> MLP Neural Network

<sup>۸</sup> Random Forest

<sup>۱</sup> SVM-RBF

<sup>۲</sup> Linear SVM

<sup>۳</sup> k-NN

<sup>۴</sup> Decision Tree

<sup>۵</sup> Cascade Forward

in International Workshop on Recent Advances in Intrusion Detection, Springer, pp. 109-129, 2014.

- [10] K. N. Khasawneh, M. Ozsoy, C. Donovick, N. Abu-Ghazaleh, and D. Ponomarev, "Ensemble learning for low-level hardware-supported malware detection," in International Symposium on Recent Advances in Intrusion Detection, Springer, pp. 3-25, 2015.
- [11] Y. Fan, Y. Ye, and L. Chen, "Malicious sequential pattern mining for automatic malware detection," Expert Systems with Applications, vol. 52, pp. 16-25, 2016.
- [12] M. Ozsoy, K. N. Khasawneh, C. Donovick, I. Gorelik, N. Abu-Ghazaleh, and D. Ponomarev, "Hardware-based malware detection using low-level architectural features," IEEE Transactions on Computers, vol. 65, no. 11, pp. 3332-3344, 2016.
- [13] S. Huda, J. Abawajy, M. Alazab, M. Abdollahian, R. Islam, and J. Yearwood, "Hybrids of support vector machine wrapper and filter based framework for malware detection," Future Generation Computer Systems, vol. 55, pp. 376-390, 2016.
- [14] M. Imran, M. T. Afzal, and M. A. Qadir, "Malware classification using dynamic features and Hidden Markov Model," Journal of Intelligent & Fuzzy Systems, vol. 31, no. 2, pp. 837-847, 2016.
- [15] S. S. Hansen, T. M. T. Larsen, M. Stevanovic, and J. M. Pedersen, "An approach for detection and family classification of malware based on behavioral analysis," in 2016 International conference on computing, networking and communications (ICNC), IEEE, pp. 1-5, 2016.
- [16] B. M. Khammas, A. Monemi, I. Ismail, S. M. Nor, and M. Marsono, "Metamorphic malware detection based on support vector machine classification of malware sub-signatures," Telkomnika (Telecommunication Computing Electronics and Control), vol. 14, no. 3, 2016.
- [17] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," arXiv preprint arXiv:1606.04435, 2016.
- [18] N. Patel, A. Sasan, and H. Homayoun, "Analyzing hardware based malware detectors," in 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), IEEE, pp. 1-6, 2017.
- [19] B. Singh, D. Evtushkin, J. Elwell, R. Riley, and I. Cervasato, "On the detection of kernel-level rootkits using hardware performance counters," in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 483-493, 2017.
- [20] H. Sayadi, N. Patel, S. M. PD, A. Sasan, S. Rafatirad, and H. Homayoun, "Ensemble learning for effective run-time hardware-based malware detection: A comprehensive analysis and classification," in 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), IEEE, pp. 1-6, 2018.
- [21] Z.-U. Rehman et al., "Machine learning-assisted signature and heuristic-based detection of malwares in Android devices," Computers & Electrical Engineering, vol. 69, pp. 828-841, 2018.
- [22] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," The bulletin of mathematical biophysics, vol. 5, no. 4, pp. 115-133, 1943.
- [23] Z. Soltani and A. Jafarian, "A new artificial neural networks approach for diagnosing diabetes disease type II," International Journal of Advanced Computer Science and Applications, vol. 7, no. 6, pp. 89-94, 2016.

## ۵- نتیجه گیری

رشد چشمگیر بدافزارها در سال‌های اخیر موجب شده است که به‌کارگیری سامانه‌های هوشمند تشخیص بدافزار به یک ضرورت تبدیل شود. این مسئله سبب شده است که حوزه تشخیص بدافزار به یک حوزه جذاب برای محققین هوش مصنوعی، داده‌کاوی و یادگیری ماشین تبدیل گردد در این مقاله روشی هوشمند جهت تشخیص بدافزارها با استفاده از ویژگی‌های عملکردی شمارنده‌های سخت‌افزاری و شبکه عصبی MLP بهینه‌شده به‌وسیله الگوریتم سنجاقک ارائه شد. با توجه به قابلیت تفکیک‌پذیری بسیار بالای ویژگی‌های رفتاری شمارنده‌های سخت‌افزاری و نیز کارایی مناسب دسته‌بند به‌کار گرفته شده، سامانه پیشنهادی توانست با دقت بیش از ۸۶ درصد فایل‌های آلوده به بدافزار را از فایل‌های سالم تشخیص دهد. نتایج حاصل از مقایسه‌ها نشان می‌دهد روش پیشنهادی در مقایسه با بهترین روش پیشین بیش از ۹ درصد بهبود دارد.

## ۶- مراجع

- [1] S. Parsa and S. Khoshruy, "A new method for gradual detection of environmental conditions and resources required by smart malware," Journal of Electrical & Cyber Defence, vol. 6, no. 4, pp. 33-44, 2019.
- [2] G. Laurenza, D. Ucci, L. Aniello, and R. Baldoni, "An architecture for semi-automatic collaborative malware analysis for cis," in 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W), IEEE, pp. 137-142, 2016.
- [3] H. Ali Alatwi, T. Oh, E. Fokoue, and B. Stackpole, "Android malware detection using category-based machine learning classifiers," in Proceedings of the 17th Annual Conference on Information Technology Education, pp. 54-59, 2016.
- [4] Z. Bazrafshan, H. Hashemi, S. M. H. Fard, and A. Hamzeh, "A survey on heuristic malware detection techniques," in The 5th Conference on Information and Knowledge Technology, IEEE, pp. 113-120, 2013.
- [5] N. Milosevic, A. Dehghantanha, and K.-K. R. Choo, "Machine learning aided Android malware classification," Computers & Electrical Engineering, vol. 61, pp. 266-274, 2017.
- [6] S. Parsa and A. Gooran Oorimi, "An Optimal and Transparent Framework for Automatic Analysis of Malware," ADST Journal, vol. 7, no. 1, pp. 71-80, 2016.
- [7] H. Hashemi, A. Azmoodeh, A. Hamzeh, and S. Hashemi, "Graph embedding as a new approach for unknown malware detection," Journal of Computer Virology and Hacking Techniques, vol. 13, no. 3, pp. 153-166, 2017.
- [8] K. Pal and J. Verma, "A Survey on Anomaly Based Malware Detection and Demolition in False Alarm Rate," Available at SSRN 2652104, 2015.
- [9] A. Tang, S. Sethumadhavan, and S. J. Stolfo, "Unsupervised anomaly-based malware detection using hardware features,"



[26] S. Y. Yerima and S. Sezer, "Droidfusion: A novel multilevel classifier fusion approach for android malware detection," IEEE transactions on cybernetics, vol. 49, no. 2, pp. 453-466, 2018.

[24] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," Neural Computing and Applications, vol. 27, no. 4, pp. 1053-1073, 2016.

[25] J. Han, J. Pei, and M. Kamber, Data mining: concepts and techniques, Elsevier, 2011.