نهمین کنفرانس ملی مباحث نوین در کامپیوتر و فناوری اطلاعات
9th National Conference on Advanced Topics in Computer and Information Technology
۲۰ آذر ماه ۱۴۰۴

استفاده از الگوریتم گرگ خاکستری جهت بهبود تشخیص انواع بدافزار در سیستم‌های رایانه‌ای

سمیرا حسینی [۱] – علیرضا صفائی [۲]

(۱) موسسه آموزش عالی پاسارگاد، شیراز، ایران
samira.hosseini۲@gmail.com
(۲) گروه کامپیوتر، واحد ماهشهر، دانشگاه آزاد اسلامی، ماهشهر، ایران
alisafasafaei@yahoo.com

چکیده: افزایش بد افزارها درسال‌های اخیر به یک نگرانی جدی تبدیل شده است. و همچنین رفتار و عملکرد بدافزارهای جدید به شدت پیچیده و مبهم شده است. از آنجایی که یادگیری ماشین به عنوان شاخه‌های اصلی هوش مصنوعی،نقش بسیار مهمی در تحلیل و تشخیص بدافزارها دارد، لذا با استفاده از شبکه عصبی پیشرو بهبود یافته بوسیله الگوریتم گرگ خاکستری به تشخیص انواع بدافزارهای رایانه‌ای با دقت بالایی پرداخته شده است. فازهای اصلی شکار در الگوریتم پیشنهادی به این صورت است،که اول ردگیری و نزدیک شدن به شکار است. سپس تعقیب و محاصره و در آخر حمله نهایی است. روش پیشنهادی مبتنی بر تحلیل داده‌ها با استفاده از ترکیبی از الگوریتم‌های داده‌کاوی و تکنیک‌های فراابتکاری طراحی شده است. تعداد تکرار آموزش شبکه عصبی پیشنهادی برابر با ۱۰۰۰ تکرار است. بررسی نتایج پیاده‌سازی نشان می‌دهد که به‌کارگیری این الگوریتم در نسخه‌ی بهبودیافته‌ی روش پیشنهادی، موجب ارتقای دقت تشخیص فایل‌های آلوده از فایل‌های سالم در مجموعه‌ی داده‌های آموزشی به میزان ۲،۸۵ درصد نسبت به نسخه‌ی اولیه گردیده است. بر اساس نتایج حاصل از اجرای روش پیشنهادی و مقایسه آن با پژوهش‌های پیشین؛ می‌توان نتیجه گرفت که این روش از کارایی مناسبی در شناسایی فایل‌های آلوده به بدافزار برخوردار بوده و توانسته است؛ فایل‌های سالم را با دقت بالا از نمونه‌های آلوده تفکیک و طبقه‌بندی نماید.

کلمات کلیدی: تشخیص بدافزار، هوش مصنوعی،تحلیل رفتار، یادگیری ماشین، امنیت سایبری، داده کاوی، الگوریتم بهینه‌سازی گرگ خاکستری

۱ – مقدمه

بدافزار یا نرم‌افزار مخرب، به برنامه‌هایی گفته می‌شود که بدون اجازه کاربر، عملکردهای ناخواسته یا مخرب را بر روی سیستم انجام می‌دهند. و می‌توانند علاوه بر اختلال در فعالیت‌های سیستم و منابع آن، با سرقت غیرمجاز اطلاعات شخصی، هزینه‌های مالی قابل‌توجهی برای کاربران ایجاد کنند. روش‌های تشخیص بدافزار به منظور شناسایی و جلوگیری از نفوذ بدافزارها به سیستم‌های کامپیوتری توسعه یافته‌اند و می‌توانند از آسیب به داده‌ها و سازش امنیتی سیستم‌ها جلوگیری کنند. به طور کلی، تکنیک‌های تشخیص بدافزار را می‌توان در دو دسته اصلی طبقه‌بندی کرد:۱- تشخیص مبتنی بر امضا ۲-تشخیص مبتنی بر ناهنجاری

هریک از روش های تشخیص بدافزار خود شامل روش های پویا روش های ایستا و روش‌های ترکیب هستند. روش مبتنی بر امضا بر تحلیل کدهای مشکوک و استخراج ویژگی‌های خاص نرم‌افزارهای مخرب استوار است. هدف اصلی این روش، شناسایی توالی‌های بایتی منحصر به فرد در کدها و استفاده از آن‌ها به‌عنوان امضا برای تشخیص بدافزارها است. همچنین، فرآیند شناسایی شامل جستجوی این امضاها در فایل‌های مشکوک می‌باشد. تجزیه و تحلیل بدافزار فرایندی چندمرحله‌ای و پویا است که با هدف شناسایی رفتار، عملکرد و اهداف نرم‌افزارهای مخرب انجام می‌شود. این فرایند به تحلیل‌گران کمک می‌کند تا دریابند بدافزار چگونه عمل می‌کند، چه نوع تهدیدی ایجاد می‌کند و چه روش‌هایی برای خنثی‌سازی آن مؤثر است. در واقع، هدف اصلی تحلیل بدافزار

دستیابی به درکی جامع از منطق عملکردی و سازوکار انتشار آن برای طراحی راهکارهای مقابله‌ای است. به منظور پیشگیری از بروز حملات سایبری و کاهش تأثیرات آن بر شبکه‌ها و سیستم‌های اطلاعاتی، بررسی و شناسایی رفتار بدافزارها اهمیت ویژه‌ای دارد. به طور کلی، دو رویکرد اصلی برای تحلیل بدافزار وجود دارد که توسط متخصصان امنیتی به کار گرفته می‌شود: تجزیه و تحلیل ایستا و تجزیه و تحلیل پویا. در روش‌های تحلیل ایستا (Static Analysis)، ارزیابی و شناسایی بدافزار بدون نیاز به اجرای واقعی فایل مخرب انجام می‌شود. در این رویکرد، با بهره‌گیری از ابزارهای مهندسی معکوس و تکنیک‌های مرتبط، ساختار داخلی و نحوه نگارش کد برنامه مورد بررسی قرار می‌گیرد. ابزارهایی مانند دی‌اسمبلر (Disassembler)، دیباگر (Debugger) و تحلیل‌گر باینری از جمله ابزارهای کلیدی در این فرآیند هستند. هدف تحلیل ایستا، استخراج اطلاعات دقیق از سطح کد منبع یا باینری برنامه و شناسایی الگوهای رفتاری مشکوک است، بدون آنکه برنامه در محیط واقعی اجرا شود. از طریق این روش می‌توان ویژگی‌هایی نظیر جریان کنترل برنامه، توابع سیستمی فراخوانی‌شده، ساختار داده‌ها و وابستگی‌های درونی را شناسایی کرد. با وجود دقت بالای این رویکرد، یکی از محدودیت‌های مهم آن وابستگی به دسترسی مستقیم به ساختار کد بدافزار است. بسیاری از بدافزارهای مدرن برای جلوگیری از تحلیل، از تکنیک‌هایی مانند رمزنگاری، فشرده‌سازی یا مبهم‌سازی کد (Code Obfuscation) استفاده می‌کنند. این امر موجب می‌شود تحلیل ایستا در مواجهه با چنین بدافزارهایی کارایی خود را از دست بدهد. به همین دلیل، برای جبران این ضعف و بررسی رفتار اجرایی بدافزار، روش‌های تحلیل پویا (Dynamic Analysis) توسعه یافته‌اند. الگوریتم گرگ خاکستری روش پیشنهادی که مدل ریاضی این الگوریتم شامل ساختار حکومت اجتماعی، محاصره و دور زدن شکار، شناسایی موقعیت طعمه، ردگیری و اجرای حمله گروهی، در قالب مدلی ریاضی و قابل شبیه‌سازی توصیف می‌گردد.در فرآیند مدلسازی ریاضی رفتار اجتماعی گرگ‌های خاکستری در طراحی الگوریتم GWO (Grey Wolf Optimizer)، بهترین پاسخ به‌دست‌آمده در هر تکرار به‌عنوان گرگ آلفا (α) در نظر گرفته می‌شود. دومین و سومین پاسخ‌های برتر به‌ترتیب به‌عنوان گرگ بتا (β) و دلتا (δ) شناخته می‌شوند، در حالی که سایر پاسخ‌ها نقش گرگ‌های امگا (ω) را ایفا می‌کنند. بنابراین، در روند بهینه‌سازی، هدایت جستجو عمدتاً توسط سه گرگ α، β و δ انجام شده و سایر اعضا با تبعیت از آنها مسیر جستجو را دنبال می‌کنند. ارائه یک روش نوین برای تشخیص فایل‌های آلوده از سالم به گونه‌ای است؛ که انواع بدافزارها در فایل‌ها شناسایی شوند. این روش مبتنی بر تحلیل داده‌ها با استفاده از ترکیبی از الگوریتم‌های داده‌کاوی و تکنیک‌های فراابتکاری طراحی شده است.

روش پیشنهادی در این تحقیق قادر است که فایل های آلوده به انواع مختلف بدافزارها را به صورت کاملاً قابل اعتماد و کارآمد شناسایی کند. ایده اصلی روش پیشنهادی استفاده از الگوریتم شبکه عصبی مصنوعی برای طبقه بندی و استفاده از الگوریتم گرگ خاکستری جهت بهبود شبکه عصبی مصنوعی می‌باشد. همانطور که گفته شد در روش پیشنهادی از یک شبکه عصبی مصنوعی جهت تشخیص فایل‌های آلوده به انواع بدافزار از فایل‌های سالم استفاده خواهیم نمود. بنابراین با توجه به پارامتر های ورودی مسئله از جمله تعداد جمعیت اولیه گرگ‌ها (۱۰۰ گرگ) تعداد تکرار (۱۰۰۰ تکرار) حد بالا و حدپایین تعداد لایه‌ها (بین یک تا سه لایه) و تعداد نرون های میانی (بین یک تا ۴۰ نرون) الگوریتم پیشنهادی کار خود را آغاز می کند بنابراین الگوریتم کار خود را در ابتدا با یک لایه میانی آغاز کرده و در ادامه تعداد لایه ها به دو لایه افزایش یافته و در نهایت با سه لایه میانی اجرا خواهد شد زمانی که شبکه عصبی یا ماتریس مورد نظر دارای یک لایه میانی است؛ الگوریتم به تعداد ۱۰۰۰ مرتبه تکرار شده و در هر تکرار تمامی گرگ‌ها از مقادیر متفاوتی در محدوده بین یک تا ۵۰ جهت تعداد نرون‌های لایه میانی برخوردار هستند. بنابراین باتوجه به نتایج بدست آمده از پیاده سازی روش پیشنهادی جهت تشخیص وجود فایل‌های آلوده به بدافزار از فایل‌های سالم می توان گفت روش پیشنهادی از عملکرد خوب و بالایی در تشخیص فایل‌های آلوده به بدافزار برخوردار است و توانسته است با دقت بالایی فایل‌های سالم را از فایل‌های بدافزاری تشخیص و طبقه‌بندی کند. [۱،۲].

۲ – شرح مقاله

روش‌های تشخیص بدافزار به منظور شناسایی و جلوگیری از نفوذ بدافزارها به سیستم‌های کامپیوتری توسعه یافته‌اند و می‌توانند از آسیب به داده‌ها و سازش امنیتی سیستم‌ها جلوگیری کنند. به طور کلی، تکنیک‌های تشخیص بدافزار را می‌توان در دو دسته اصلی طبقه‌بندی کرد: تشخیص مبتنی بر امضا (Signature-Based Detection)
تشخیص مبتنی بر ناهنجاری (Anomaly-Based etection).

هریک از روش‌های تشخیص بدافزار خود شامل روش‌های پویا روش‌های ایستا است. روش مبتنی بر امضا (Signature-Based Detection) بر تحلیل کدهای مشکوک و استخراج ویژگی‌های خاص نرم‌افزارهای مخرب استوار است. هدف اصلی این روش، شناسایی توالی‌های بایتی منحصر به فرد در کدها و استفاده از آن‌ها به‌عنوان امضا برای تشخیص بدافزارهاست. همچنین، فرآیند شناسایی شامل جستجوی این امضاها در فایل‌های مشکوک می‌باشد. بیشتر نرم‌افزارهای ضدبدافزار تجاری از مجموعه‌ای از امضاها برای شناسایی برنامه‌های مخرب استفاده می‌کنند؛ فایل‌های مشکوک با این توالی‌ها مقایسه شده و در صورتی که امضای مربوطه در پایگاه داده موجود نباشد، فایل مورد نظر به‌عنوان سالم تلقی می‌شود.

با این حال، روش مبتنی بر امضا دارای محدودیت‌هایی است. یکی از مهم‌ترین چالش‌ها، نیاز به دخالت انسان برای به‌روزرسانی پایگاه داده با امضاهای جدید است. [۳،۴،۵،۶].

روش‌های تشخیص مبتنی بر ناهنجاری معمولاً شامل دو فاز مجزا هستند: فاز آموزش و فاز تشخیص (نظارت). در فاز آموزش، سامانه‌ی کشف‌کننده رفتار «نرمال» را می‌آموزد؛ این یادگیری می‌تواند از داده‌های مربوط به میزبان، برنامه‌های تحت نظارت یا ترکیبی از آن‌ها صورت پذیرد. در مرحله‌ی بعد (فاز تشخیص)، سامانه رفتار برنامه‌ی در حال اجرا را با الگوهای یادگرفته‌شده مقایسه می‌کند تا هرگونه انحراف یا ناپایداری احتمالی را شناسایی نماید. یکی از مزایای برجسته‌ی این رویکرد توانایی تشخیص حملات «صفر روز» (zero day) است، زیرا

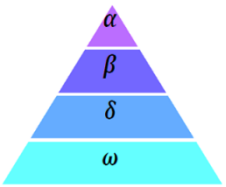
سیستم بر اساس رفتار غیرعادی عمل می‌کند نه امضای شناخته‌شده. با این حال، این روش محدودیت‌هایی نیز دارد: نرخ مثبت کاذب معمولاً بالاتر است و تعیین مجموعه‌ی مناسب ویژگی‌ها برای آموزش (feature selection) می‌تواند پیچیده و زمان‌بر باشد. علاوه بر این، از آن‌جا که فاز آموزش هرگز قادر به دیدن تمامی حالات ممکن رفتارهای آینده نیست، احتمال بروز خطاهای مثبت کاذب همچنان وجود دارد. در نتیجه، کارایی نهایی سامانه‌ی مبتنی بر ناهنجاری به کیفیت داده‌های آموزش، انتخاب ویژگی‌ها و تنظیم آستانه‌های تشخیصی وابسته است.[۵،۷].

تجزیه و تحلیل بدافزار فرایندی چندمرحله‌ای و پویا است که با هدف شناسایی رفتار، عملکرد و اهداف نرم‌افزارهای مخرب انجام می‌شود. این فرایند به تحلیل‌گران کمک می‌کند تا دریابند بدافزار چگونه عمل می‌کند، چه نوع تهدیدی ایجاد می‌کند و چه روش‌هایی برای خنثی‌سازی آن مؤثر است. در واقع، هدف اصلی تحلیل بدافزار دستیابی به درکی جامع از منطق عملکردی و سازوکار انتشار آن برای طراحی راهکارهای مقابله‌ای است. به منظور پیشگیری از بروز حملات سایبری و کاهش تأثیرات آن بر شبکه‌ها و سیستم‌های اطلاعاتی، بررسی و شناسایی رفتار بدافزارها اهمیت ویژه‌ای دارد. به طور کلی، دو رویکرد اصلی برای تحلیل بدافزار وجود دارد که توسط متخصصان امنیتی به کار گرفته می‌شود: تجزیه و تحلیل ایستا و تجزیه و تحلیل پویا. در روش‌های تحلیل ایستا (Static Analysis)، ارزیابی و شناسایی بدافزار بدون نیاز به اجرای واقعی فایل مخرب انجام می‌شود. در این رویکرد، با بهره‌گیری از ابزارهای مهندسی معکوس و تکنیک‌های مرتبط، ساختار داخلی و نحوه نگارش کد برنامه مورد بررسی قرار می‌گیرد. ابزارهایی مانند دی‌اسمبلر (Disassembler)، دیباگر (Debugger) و تحلیل‌گر باینری از جمله ابزارهای کلیدی در این فرآیند هستند. هدف تحلیل ایستا، استخراج اطلاعات دقیق از سطح کد منبع یا باینری برنامه و شناسایی الگوهای رفتاری مشکوک است، بدون آنکه برنامه در محیط واقعی اجرا شود. از طریق این روش می‌توان ویژگی‌هایی نظیر جریان کنترل برنامه، توابع سیستمی فراخوانی‌شده، ساختار داده‌ها و وابستگی‌های درونی را شناسایی کرد. با وجود دقت بالای این رویکرد، یکی از محدودیت‌های مهم آن وابستگی به دسترسی مستقیم به ساختار کد بدافزار است. تحلیل پویا با استفاده از روش‌ها و فنون مختلفی انجام می گیرد .

الگوریتم گرگ خاکستری:

الگوریتم گرگ خاکستری (Grey Wolf Optimizer) بر اساس رفتار شکار دسته‌جمعی گرگ‌ها طراحی گردیده است[۱۱].

گرگ‌های خاکستری در بالای زنجیره غذایی قرار دارند و معمولاً به صورت گروهی زندگی می‌کنند که میانگین اندازه گروه‌ها بین ۵ تا ۱۲ عضو است. این گروه‌ها دارای سلسله مراتب اجتماعی مشخص و سختگیرانه‌ای هستند که بر اساس آن، گرگ آلفا به عنوان رهبر گروه شناخته می‌شود و تصمیم‌گیری‌های کلیدی مانند زمان و مکان شکار، جای خواب و حرکت گروه را هدایت می‌کند. در سطح بعدی، گرگ‌های بتا قرار دارند که نقش مشاور و معاون آلفا را ایفا می‌کنند و همچنین مدیریت گروه را تا حدی بر عهده دارند. پایین‌ترین سطح سلسله مراتب، گرگ‌های امگا هستند که غالباً آخرین افرادی هستند که اجازه خوردن غذا دارند و نقش «قربانی» را برای دیگر اعضا ایفا می‌کنند. اعضایی که در هیچ یک از این سه گروه قرار نگرفته باشند، تحت عنوان دلتا شناخته می‌شوند؛ دلتاها از آلفاها و بتاها تبعیت کرده و بر امگاها سلطه دارند و شامل پیشاهنگ‌ها و دیده‌بان‌ها نیز می‌شوند. رفتار اجتماعی گرگ‌ها به ویژه شکار دسته‌جمعی، پایه اصلی الگوریتم گرگ خاکستری را تشکیل می‌دهد و در شکل ۴، سلسله مراتب اجتماعی و عملکرد گروهی گرگ‌ها نمایش داده شده است.[۸،۹،۱۰،۱۱].



شکل(۱): سلسله مراتب گرگ خاکستری[۱۱].

فازهای اصلی شکار در الگوریتم گرگ خاکستری به ترتیب زیر انجام می‌شوند:

۱.ردگیری و نزدیک شدن به شکار: گرگ‌ها ابتدا شکار را شناسایی کرده، آن را وادار به فرار می‌کنند و به تدریج به آن نزدیک می‌شوند.

۲.تعقیب و محاصره: در این مرحله، گرگ‌ها شکار را دنبال کرده، با ایجاد فشار و محاصره، حرکت آن را محدود می‌کنند تا شکار از حرکت بازایستد .

۳.حمله نهایی: در پایان، گرگ‌ها به صورت هماهنگ به شکار حمله کرده و آن را شکار می‌کنند.



شکل (۲): رفتار شکار گرگ‌های خاکستری[۱۱].

مدل ریاضی الگوریتم گرگ خاکستری در این بخش، الگوی رفتاری گرگ‌های خاکستری شامل ساختار حکومت اجتماعی، محاصره و دور زدن شکار، شناسایی موقعیت طعمه، ردگیری و اجرای حمله گروهی، درقالب مدلی ریاضی و قابل شبیه‌سازی توصیف می‌گردد.در فرآیند مدلسازی ریاضی رفتار اجتماعی گرگ‌های خاکستری در طراحی

الگوریتم GWO (Grey Wolf Optimizer)، بهترین پاسخ به دست‌آمده در هر تکرار به‌عنوان گرگ آلفا (α) در نظر گرفته می‌شود. دومین و سومین پاسخ‌های برتر به‌ترتیب به‌عنوان گرگ بتا (β) و دلتا (δ) شناخته می‌شوند، در حالی که سایر پاسخ‌ها نقش گرگ‌های امگا (ω) را ایفا می‌کنند. بنابراین، در روند بهینه‌سازی، هدایت جستجو عمدتاً توسط سه گرگ α، β و δ انجام شده و سایر اعضا با تبعیت از آن‌ها مسیر جستجو را دنبال می‌کنند.

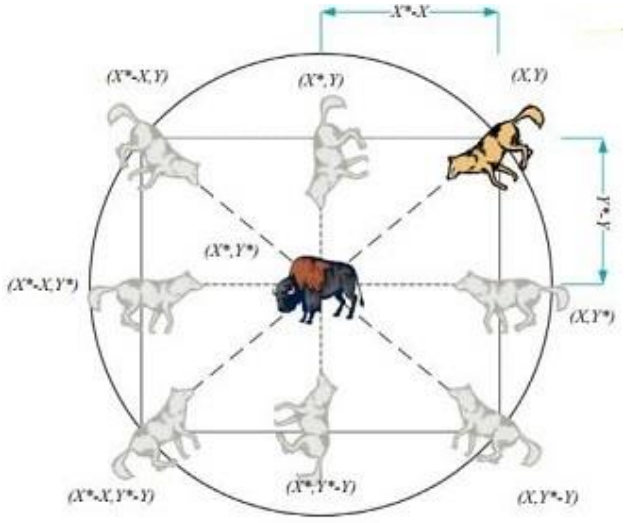
در فرآیند شکار، گرگ‌های خاکستری با حرکت دایره‌ای و محاصره گونه طعمه را احاطه می‌کنند. جهت نمایش ریاضی این رفتار محاصره‌ای و نزدیک شدن تدریجی به شکار، فرمول(۱) مورد استفاده قرار گرفته‌اند.

$$\vec{D} = \left|\vec{C}\vec{X}_P(t) - \vec{X}(t)\right| \qquad (۱)$$
$$\vec{X}(t+۱) = \vec{X}_P(t) - \vec{A}\vec{D}$$

که t نشان دهنده تعداد تکرار فعلی $\vec{A}$ و $\vec{C}$ ضریب بردار $\vec{X}_P$ بردار موقعیت شکار و $\vec{X}$ نشان دهنده بردار موقعیت یک گرگ است. بردار های $\vec{A}$ و $\vec{C}$ مطابق با معادلات (۲) محاسبه می شوند:

$$\vec{A} = ۲\vec{a}\vec{r}_۱ - \vec{a}$$
$$\vec{C} = ۲\vec{r}_۲ \qquad (۲)$$

مولفه‌های $\vec{a}$ از ۲ تا ۰ در طول تکرار ها بطور خطی کاهش می‌یابند و $\vec{r}_۱$ و $\vec{r}_۲$ بردار های تصادفی در بازهٔ [۱و۰] می‌باشند. نتایج معادلات به همراه یک بردار مکان دو بعدی تعدادی همسایگی احتمالی در شکل(۳) نشان داده شده‌اند.



شکل(۳): بردار های دوبعدی و موقعیت احتمالی بعدی آنها[۱۱].

مطابق شکل(۳)، گرگ خاکستری که در موقعیت (X,Y) قرار دارد، می‌تواند با توجه به موقعیت طعمه در (X*,Y*) جایگاه خود را تغییر دهد. موقعیت‌های مختلف اطراف بهترین عامل را می‌توان بر اساس مکان فعلی و با تنظیم بردارهای $\vec{A}$ و $\vec{C}$ تعیین کرد. برای نمونه، نقطه‌ی (X*-X,Y*) از طریق مقداردهی $\vec{A}$=(۱،۰) و $\vec{C}$=(۱،۱) به‌دست می‌آید. موقعیت‌های احتمالی به‌روزشده برای یک گرگ خاکستری در فضای سه‌بعدی در شکل (۳) نمایش داده شده است. باید توجه داشت که بردارهای تصادفی $r^۱$ و $r^۲$ این امکان را فراهم می‌کنند تا گرگ بتواند به هر موقعیتی میان نقاط نمایش‌داده‌شده است. در نتیجه، گرگ خاکستری قادر است با بهره‌گیری از معادلات (۳) به‌صورت تصادفی موقعیت خود را در محدوده‌ای که طعمه در آن قرار دارد تغییر دهد. این مفهوم را می‌توان به فضای جستجوی n بعدی نیز تعمیم داد؛ در این حالت، گرگ‌های خاکستری در اطراف بهترین راه‌حل کشف‌شده در فضایی با ابعاد بیشتر از مکعب حرکت می‌کنند.

گرگ‌های خاکستری قادرند موقعیت طعمه را شناسایی کرده و آن را محاصره کنند. فرآیند شکار معمولاً تحت هدایت گرگ آلفا انجام می‌شود، هرچند گاهی گرگ‌های بتا و دلتا نیز در این فرایند مشارکت دارند. از آنجا که در فضای جستجوی کلی، موقعیت دقیق شکار یا همان پاسخ بهینه مشخص نیست، برای مدل‌سازی ریاضی رفتار شکار، فرض می‌شود که آلفا (به‌عنوان بهترین راه‌حل)، بتا و دلتا از آگاهی نسبی نسبت به موقعیت احتمالی طعمه برخوردارند. بر این اساس، سه راه‌حل برتر حاصل از جستجو ذخیره شده و سایر گرگ‌ها (امگاها) موقعیت خود را بر مبنای موقعیت این سه عامل برتر به‌روزرسانی می‌کنند. فرمول (۳) مشخص می کند.

$$\vec{D}_a = \left|\vec{C}_۱\vec{X}_a - \vec{X}\right| \vec{D}_\beta = \left|\vec{C}_۲\vec{X}_\beta - \vec{X}\right| \vec{D}_\delta = \left|\vec{C}_۳\vec{X}_\delta - \vec{X}\right|$$
$$\vec{X}_۱ = \vec{X}_a - \vec{A}_۱(\vec{D}_a)\vec{X}_۲ = \vec{X}_\beta - \vec{A}_۲(\vec{D}_\beta), \vec{X}_۳ = \vec{X}_\delta - \vec{A}_۳(\vec{D}_\delta) \qquad \textbf{(۳)}$$
$$\vec{X}(\vec{t}+۱) = \frac{\vec{X}_۱ + \vec{X}_۲ + \vec{X}_۳}{۳}$$

یک عامل جستجو موقعیت خود را بر اساس جایگاه‌های آلفا، بتا و دلتا در فضای جستجوی دو بعدی به‌روزرسانی می‌کند. موقعیت نهایی عامل می‌تواند به‌طور تصادفی در محدوده‌ای اطراف موقعیت‌های آلفا، بتا و دلتا قرار گیرد. به عبارت دیگر، آلفا، بتا و دلتا موقعیت شکار را تخمین می‌زنند و سایر گرگ‌ها موقعیت خود را به‌صورت تصادفی در اطراف شکار تنظیم می‌کنند. گرگ‌های خاکستری حمله خود را زمانی تکمیل می‌کنند که طعمه دیگر حرکتی نداشته باشد. برای مدل‌سازی ریاضی موقعیت طعمه، مقدار α\alphaکاهش داده می‌شود. توجه کنید که محدوده تغییرات $\vec{A}$ بوسیله $\vec{a}$ کاهش یافته است به عبارت دیگر $\vec{A}$ یک مقدار تصادفی در فاصله [-2a ,2a] است؛ که a از ۲ به ۰ تحت مسیر تکرارها کاهش می یابد. زمانی که بردار تصادفی A در محدوده [-۱, ۱] قرار دارد، موقعیت بعدی عامل جستجو می‌تواند در هر نقطه‌ای بین جایگاه فعلی خود و موقعیت طعمه قرار گیرد. با توجه به عملگرهایی که تاکنون معرفی

شده‌اند، الگوریتم گرگ خاکستری این امکان را به عوامل جستجو می‌دهد تا موقعیت خود را بر اساس جایگاه آلفا، بتا و دلتا به‌روزرسانی کرده و به طعمه نزدیک شوند. با این وجود، این الگوریتم ممکن است در حداقل‌های محلی گرفتار شود.

بنابراین استفاده از عملگرهای تکمیلی برای جلوگیری از گرفتار شدن در حداقل‌های محلی ضروری است. اگرچه مکانیسم محاصره تا حدی قابلیت شناسایی موقعیت را فراهم می‌کند، الگوریتم برای افزایش توانایی اکتشاف نیازمند عملگرهای اضافی است. گرگ‌های خاکستری اساساً براساس جایگاه آلفا، بتا و دلتا به جستجو می‌پردازند. یکی دیگر از اجزای مؤثر در الگوریتم گرگ خاکستری که بر فرآیند شناسایی تأثیر دارد، بردار C است. همان‌طور که در معادله (۳) مشاهده می‌شود، بردار C شامل مقادیر تصادفی در بازه [۰, ۲] است. این بردار برای شکار نقش وزن‌های تصادفی را ایفا می‌کند و میزان تأثیر موقعیت طعمه در تعیین فاصله در معادله (۱) را می‌تواند افزایش دهد (C>۱) یا کاهش دهد (C<۱). همچنین این عامل باعث می‌شود الگوریتم گرگ خاکستری رفتار تصادفی بیشتری در طول بهینه‌سازی داشته باشد که منجر به افزایش قابلیت اکتشاف و کاهش احتمال گرفتار شدن در حداقل‌های محلی می‌شود.

در الگوریتم گرگ خاکستری، فرایند جستجو با ایجاد یک جمعیت تصادفی از گرگ‌های خاکستری که به‌عنوان راه‌حل‌های کاندید عمل می‌کنند، آغاز می‌شود. در طول تکرارها، گرگ‌های آلفا، بتا و دلتا موقعیت‌های احتمالی طعمه را برآورد می‌کنند و سایر راه‌حل‌های کاندید فاصله خود را نسبت به طعمه به‌روزرسانی می‌نمایند. پارامتر aاز مقدار ۲ به صفر کاهش می‌یابد تا فرآیند شناسایی و حمله به طعمه تقویت شود. زمانی که |A|>۱ باشد، راه‌حل‌های کاندید از طعمه واگرا می‌شوند و هنگامی که |A|<۱ باشد، به سمت طعمه همگرا می‌گردند.

ارائه یک روش نوین برای تشخیص فایل‌های آلوده از سالم است، به گونه‌ای که انواع بدافزارها در فایل‌ها شناسایی شوند. این روش مبتنی بر تحلیل داده‌ها با استفاده از ترکیبی از الگوریتم‌های داده‌کاوی و تکنیک‌های فراابتکاری طراحی شده است.

روش پیشنهادی قادر است که فایل های آلوده به انواع مختلف بدافزارها را به صورت کاملاً قابل اعتماد و کارآمد شناسایی کند. ایده اصلی روش پیشنهادی استفاده از الگوریتم شبکه عصبی مصنوعی برای طبقه بندی و استفاده از الگوریتم گرگ خاکستری جهت بهبود شبکه عصبی مصنوعی می‌باشد همانطور که گفته شد در روش پیشنهادی از یک شبکه عصبی مصنوعی جهت تشخیص فایل های آلوده به انواع بدافزار از فایل های سالم استفاده خواهیم نمود. برای شبکه عصبی مصنوعی از الگوریتم گرگ خاکستری استفاده کنیم روند تعیین تعداد لایه‌ها و نرون های میانی توسط الگوریتم گرگ خاکستری بدین‌صورت خواهد بود که هر گرگ در جمعیت اولیه بصورت برداریست که حاوی تعداد لایه‌ها و نرون های میانی شبکه عصبی می باشد به عنوان مثال در صورتی‌که جمعیت اولیه شامل ۱۰۰ گرگ باشد ماتریسی تشکیل خواهد شد؛ که دارای ۱۰۰ سطر (به تعداد جمعیت اولیه گرگ‌ها) است و تعداد ستون های ماتریس برابر با تعداد لایه‌های میانی شبکه عصبی خواهد بود همچنین مقادیر موجود در هر خانه ماتریس معادل با تعداد نرون‌های لایه میانی می‌باشد. بنابراین با توجه به پارامتر های ورودی مسئله از جمله تعداد جمعیت اولیه گرگ‌ها (۱۰۰ گرگ) تعداد تکرار (۱۰۰۰ تکرار) حد بالا و حدپایین تعداد لایه‌ها (بین یک تا سه لایه) و تعداد نرون های میانی (بین یک تا ۴۰ نرون) الگوریتم پیشنهادی کار خود را آغاز می‌کند. بنابراین الگوریتم کار خود را در ابتدا با یک لایه میانی آغاز کرده و در ادامه تعداد لایه‌ها به دو لایه افزایش یافته و در نهایت با سه لایه میانی اجرا خواهد شد زمانی که شبکه عصبی یا ماتریس مورد نظر دارای یک لایه میانی است الگوریتم به تعداد ۱۰۰۰ مرتبه تکرار شده و در هر تکرار تمامی گرگ‌ها از مقادیر متفاوتی در محدوده بین یک تا ۵۰ جهت تعداد نرون‌ها برخوردار هستند. پس اتمام این مرحله یعنی رسیدن به تکرار ۱۰۰۰ام الگوریتم کار خود را با سه لایه میانی شروع خواهد کرد به نحوی که ماتریس موردنظر از سه ستون تشکیل شده است. بنابراین پس از پایان ۱۰۰۰ بار تکرار در مرحله سوم میبایست به ارزیابی شبکه عصبی موردنظر پرداخت یعنی بررسی شود که از بین حالات مختلفی که اجرا شده است؛ در چه حالتی با چه تعداد لایه و نرون میانی الگوریتم از دقت بالاتری برخوردار بوده است. پس از پایان آموزش شبکه عصبی پیشنهادی شرط خاتمه بررسی می شود تا در صورت رسیدن به شرط مورد نظر آموزش آن خاتمه یافته و به ارزیابی شبکه عصبی پیشنهادی بپردازیم و در صورتی که به شرط خاتمه نرسیده باشیم الگوریتم همچنان به آموزش شبکه عصبی بپردازد در این تحقیق شرط خاتمه تعداد تکرار آموزش شبکه عصبی پیشنهادی می باشد که برابر با ۱۰۰۰ تکرار است.

۳-نتیجه

مجموعه داده مورد نظر به تعداد ۱۰۰۰ مرتبه مورد آموزش و ارزیابی قرار گرفته است. در ادامه روش پیشنهادی را توسط الگوریتم گرگ خاکستری بهبود داده و اجرا شده است. در این حالت شبکه عصبی مصنوعی بهبود یافته به تعداد ۱۰۰۰ مرتبه مورد آموزش و ارزیابی قرار گرفت که نتایج بدست آمده برای داده های آموزشی و داده های آزمایشی در جدول(۴) نشان داده شده است.

جدول (۳): مقایسه میزان درصد بهبود روش پیشنهادی در حالات بهبود یافته و بهبود نیافته است.

داده های	دقت	حساسیت	ویژگی	صحت	F-MEASUR
آموزشی	۲۸۵	۲۸۱	۲۸۹	۲۸۲	۲۸۲
آزمایشی	۶۸۱	۶۵۹	۷۰۲	۶۵۲	۶۵۶

همانگونه در جدول (۴) نشان داده شده است؛ دقت روش پیشنهادی بهبود یافته توسط الگوریتم گرگ خاکستری نسبت به حالت بهبود نیافته جهت تشخیص وجود فایل های آلوده به بدافزار از فایل های سالم برای

داده های آموزشی به میزان ۲۸۵ درصد افزایش داشته است. مقادیر معیارهای حساسیت ویژگی صحت و F-Measur به ترتیب ۲۸۱ درصد ۲۸۹ درصد ۲۸۲ درصد و ۲۸۲ درصد در حالت بهبود یافته نسبت به حالت بهبود نیافته بیشتر شده است، همچنین برای داده های آزمایشی مقادیر معیار های دقت حساسیت ویژگی صحت و F-Measur در حالت بهبود یافته نسبت به حالت بهبود نیافته به ترتیب به میزان ۶۸۱ درصد ۶۵۹ درصد ۷۰۲ درصد ۶۵۲ درصد و ۶۵۶ درصد افزایش داشته است بنابراین باتوجه به نتایج بدست آمده از پیاده سازی روش پیشنهادی جهت تشخیص وجود فایل های آلوده به بدافزار از فایل های سالم می توان گفت روش پیشنهادی از عملکرد خوب و بالایی در تشخیص فایل های آلوده به بدافزار برخوردار است و توانسته است با دقت بالایی فایل های سالم را از فایل های بدافزاری تشخیص و طبقه بندی کند.

جدول (۴): مقایسه روش پیشنهادی با برخی از کار های پیشین.

روش پیشنهاد شده توسط محققان	بیشترین دقت بدست آمده
استفاده از طبقه‌بند های بیزساده رگرسیون لجستیک و ماشین بردار پشتیبان و انتخاب ویژگی با روش جنگل تصادفی و K-Best	استفاده از رگرسیون لجستیک و انتخاب ۲۰ ویژگی بوسیله جنگل تصادفی با دقت ۹۵۵۳ درصد
۵-Interval و ۳-Interval و تکنیک گسسته‌سازی با بیز ساده	۵-Interval و تکنیک گسسته‌سازی با بیز ساده ۷۹۹۷ درصد
روش پیشنهادی ۹۸/۳۵	

باتوجه به مقایسات صورت گرفته می توان گفت روش پیشنهادی در این مقاله نسبت به مقالات [۱۲،۱۳] مورد مقایسه از دقت بالاتر و در نتیجه عملکرد بهتری در تشخیص وجود فایل‌های آلوده به بدافزار از فایل های سالم برخوردار است.

مراجع

[۱] S. Jamalpur, Y. S. Navya, P. Raja, G. Tagore, and G. R. Rao, “Dynamic malware analysis using Cuckoo Sandbox,” in *Proc. Second Int. Conf. on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, ۲۰۱۸, pp. ۱۰۵٦–۱۰٦۰.

[۲] Golbaghi, H Vahidi, M Khalilian, A (۲۰۱٦), a New Approach to Detecting Malware Malware by Static Analysis and Operational Codes

[۳] Peiravian, N (۲۰۱۳) Data Mining Heuristic-Based Malware Detection for Android

[۴] M. Damshenas, A. Dehghantanha, K. K. R. Choo, and R. Mahmud, “M۰droid: A behavior-based malware detection model for Android,” *Journal of Information Privacy and Security*, vol. ۱۱, no. ۳, pp. ۱۴۱۱۵۷, ۲۰۱۵.

[۵] Idika, N, & Mathur, A P (۲۰۰۷) A survey of malware detection techniques Purdue University, ٤۸, ۲۰۰۷-۲

[٦] Gutmann, P (۲۰۰۷) The commercial malware industry In DEFCON conference

[۷] J. Z. Kolter and M. A. Maloof, “Learning to detect and classify malicious executables in the wild,” *Journal of Machine Learning Research*, vol. ۷, no. ۱۲, ۲۰۰٦.

[۸] A. Damodaran, F. Di Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, “A comparison of static, dynamic, and hybrid analysis for malware detection,” *Journal of Computer Virology and Hacking Techniques*, vol. ۱۳, no. ۱, pp. ۱–۱۲, ۲۰۱۷.

[۹] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, “A survey on automated dynamic malware-analysis techniques and tools,” *ACM Computing Surveys (CSUR)*, vol. ٤٤, no. ۲, pp. ۱–٤۲, ۲۰۰۸.

[۱۰] Neuner, S, Van der Veen, V, Lindorfer, M, Huber, M, Merzdovnik, G, Mulazzani, M, & Weippl, E (۲۰۱٤) Enter sandbox: Android sandbox comparison arXiv preprint arXiv:۱٤۱۰۷۷٤۹

[۱۱] Mirjalili, S, Mirjalili, S M, & Lewis, A (۲۰۱٤) Grey Wolf Optimizer Adv Eng Softw ٦۹: ٤٦–٦۱

[۱۲] O. P. Samantray and S. N. Tripathy, “Knowledge-domain analysis approach for malware classification,” in *Proc. 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pp. ۱–۷, IEEE, ۲۰۲۰.

[۱۳] Anggraini, I, Kunang, Y N, & Firdaus, F (۲۰۲۰) Penerapan Naive Bayes Pada Detection Malware dengan Diskritisasi Variabel Telematika, ۱۳(۱), pp۱۱-۲۱