نشریه علمی «نوآوری‌های فناوری اطلاعات و ارتباطات کاربردی»
سال اول، شماره ۲، زمستان ۱۳۹۸؛ ص ۵۰- ۴۱

استفاده از شبکه‌های خودرمزنگار موازی به منظور تشخیص صفحات جعلی اینترنتی

نوشین امیری[۲*]، ایمان نادری[۲]

۱- کارشناسی ارشد، دانشگاه آزاد اسلامی واحد دورود، ۲- کارشناسی ارشد، دانشگاه آزاد اسلامی واحد بروجرد

(دریافت: ۱۳۹۹/۰۱/۱۳، پذیرش: ۱۳۹۹/۰۷/۰۵)

چکیده

به کمک صفحات جعلی اینترنتی تلاش می‌شود، اطلاعات محرمانه یک کاربر مانند رمز حساب‌های بانکی و گذرواژه پست الکترونیکی به سرقت برده شود. این صفحات جعلی درواقع مشابه با صفحات وب‌سایت‌های معتبر موجود مانند درگاه‌های پرداخت اینترنتی، یاهو و گوگل ساخته می‌شوند و به گونه‌ای کاربران به سمت این صفحات کشانده می‌شوند. به این نوع حمله اینترنتی، حمله فیشینگ گفته می‌شود. تشخیص برخط صفحات جعلی به کمک نرم‌افزارهای هوشمند می‌تواند جلوی به سرقت رفتن اطلاعات کاربران را گرفته و امنیت را در فضای وب افزایش دهد. در این مقاله یک روش جدید مبتنی بر شبکه‌های عصبی مصنوعی از نوع خود رمزنگار معرفی شده است. در روش پیشنهادی از دو شبکه خود رمزنگار موازی استفاده شده است که یکی از آن‌ها، با صفحات معمولی و دیگری با صفحات جعلی آموزش داده شده است. در زمان تشخیص، بر اساس بردارهای رمز شده به‌دست آمده از هر دو شبکه موازی و یک لایه شبکه عصبی مصنوعی معمولی مانند سافت مکس، نوع صفحه اینترنتی ورودی مشخص می‌شود. در کاربردهای عملی هرگاه چنین صفحه‌ای جعلی شناخته شود، به سرعت از طریق مرورگر به کاربر اخطار داده شده یا دسترسی مسدود می‌شود. نتایج آزمایش روش پیشنهادی به کمک مجموعه داده Phishing Websites و معیارهای صحت متوسط، دقت و فراخوانی، به خوبی نشان می‌دهند که شبکه‌های خود رمزنگار موازی عملکرد قوی‌تری نسبت به سایر روش‌های یادگیری ماشین در تشخیص صفحات جعلی اینترنتی دارد.

کلید واژه‌ها:

صفحات جعلی اینترنتی، حمله فیشینگ، یادگیری ماشین، شبکه‌های عصبی خود رمزنگار موازی

۱- مقدمه

امروزه با گسترش فناوری ارتباطات پست الکترونیک، شبکه‌های اجتماعی[۱]، گپ[۲] و دیگر ابزارهای موجود با وجود کاربردهای انکارناپذیر، می‌تواند مخاطرات زیادی برای کاربران مختلف داشته باشد. یکی از این خطرات وجود وب‌سایت‌های مختلفی است که در فضای مجازی وجود دارند و از آن‌ها به عنوان وب‌سایت‌های جعلی[۳] یاد می‌شود. در وب‌سایت‌های جعلی محتویات صفحه به گونه‌ای طراحی می‌شود که مشابه یک صفحه قانونی و مورد علاقه‌ی کاربران است. در اینگونه از جعل، یک وب‌سایت نقش وب‌سایت‌های دیگر را ایفاء کرده تا اطلاعات با ارزش کاربران را مورد سرقت قرار دهد. یکی از پیامدهای صفحات جعلی استفاده از آن در سرقت‌های فیشینگ[۴] است [۱].

حملات موسوم به فیشینگ به آن دسته از حملات اینترنتی گفته می‌شود که معمولاً طراحان آن‌ها به صورت ناشناس برای کشاندن کاربران به وب‌سایت‌های مورد نظرشان از پست الکترونیک استفاده می‌کنند. در اینگونه حملات، معمولاً پست‌های الکترونیکی برای کاربران ارسال می‌شود که دارای آدرس فرستنده مربوط به شرکت‌های معروف و یا بانک‌های معتبر هستند و درون آن ها نیز پیوند[۵]هایی قرار دارد که ظاهراً به همان مراکز تعلق دارند اما در حقیقت کاربر را به سوی سایت‌های مورد نظر طراحان فیشینگ هدایت می‌کنند و اطلاعات حساس نظیر کلمات عبور و یا رمز کارت‌های اعتباری کاربران را به سرقت می‌برند [۳-۲].

مقیمی و دیگران در [۲] هفت مرحله برای اجرای یک حمله فیشینگ تعریف کرده‌اند. در حقیقت فیشر پس از طراحی یک وب‌سایت جعلی مشابه با یک وب‌سایت مشروع، مراحل هفت‌گانه زیر را برای اجرای حمله در پیش می‌گیرد:

۱- ایجاد و ارسال پیوندهای بدکار: فیشرها ابتدا در مورد بنگاه تجاری هدف مطالعه کرده و شیوه به‌دست آوردن آدرس‌های پست الکترونیکی مشتریان بنگاه مورد نظر را مشخص می‌نمایند. آن‌ها غالباً از همان روش‌های جمع‌آوری آدرس و ارسال پست‌های الکترونیک انبوه[۶] استفاده می‌کنند. پس از آن‌که بنگاه تجاری که باید هویت آن جعل شود مشخص

رایانامه نویسنده پاسخگو: amirinoshin@iaud.ac.ir

[1] Social Networks
[2] Chat
[3] Fake site
[4] Phishing
[5] Link
[6] Mass Mailing

گردیـد و قربانیـان آن‌هـا نیـز شناسـایی شـدند، فیشـرها روش‌هایی را برای تحویل پیـام و جمـع‌آوری داده‌هـا ایجـاد می‌کنند. در اکثر موارد این فرآیند شامل ایجاد آدرس پست الکترونیک و یک صفحه وب است؛ سپس یـک پیـام جعلـی ارسال می‌کنند که ظاهر آن نشان می‌دهد توسط یک منبـع قابل اطمینان فرستاده شده است.

۲- دریافت پیوند بدکار توسط کـاربران: در ایـن مرحلـه کـاربر پست الکترونیک دریافت شده را باز کرده و روی پیوند بدکار درون آن کلیک می‌نماید.

۳- هدایت به سایت جعلی: پـس از کلیـک روی پیونـد بـدکار، کاربر به سایت جعلی هـدایت شـده و محتویـات آن سـایت برای وی نمایش داده می‌شود.

۴- ورود اطلاعات محرمانه در سایت جعلی: کـاربر بـا مشـاهده محتویات سایت به دلیل تشابه بسیار زیاد سـایت جعلـی بـا سایت اصلی، فریب خورده و اطلاعات شخصی خود را در آن سایت وارد می‌کند.

۵- انتقال اطلاعات کاربر: پس از اتمام ورود اطلاعـات از سـوی کاربر، اطلاعات وی از طریق همین سایت جعلی برای فیشـر ارسال می‌گردد.

۶- جعل هویت کاربر: اکنون فیشر اطلاعات حیاتی کـاربر را بـه دست آورده و به راحتی می‌تواند از این اطلاعات به هر شکل که می‌خواهد استفاده نماید.

۷- دریافـت وجـه از حسـاب کـاربر: فیشـر بـا بـه‌دسـت آوردن اطلاعات حساب بانکی قربانی به بانک وی مراجعه کرده و از حساب بانکی او پول برداشت می‌کند.

پنهان بودن الگوهای حملات و وب‌سایت‌های فیشینگ یکـی از مشکلات شناسایی اینگونه وب‌سایت‌ها می‌باشـد کـه تشـخیص اینگونه حملات را با دشواری مواجه مـی‌سـازد. یکـی از ابزارهـای مهم یادگیری ماشین[1] جهت تشخیص الگو و کشف ویژگـی‌هـای پنهان، شبکه عصبی مصنوعی[2] است که به کمک آن می‌تـوان تـا حد زیاد و با دقتی مناسب حملات فیشـینگ را شناسـایی نمـود [۴و۵].

با وجود پیچیدگی‌های موجود، صـفحات جعلـی عمـدتاً یـک سری ویژگی‌های مشترک دارند که با استخراج آن‌ها در بسـیاری از حالات می‌توان به سرعت صفحات جعلی را تشخیص داد. برخی از این ویژگی‌ها عبارتند از تشخیص دست‌کاری پیوند، مدت زمان ثبت صفحه در فضای وب، ترافیک و بازدید صفحه، آی‌پی صـفحه و دیگر ویژگی‌هـای مشـابه. در ایـن مقالـه بـا اسـتفاده از چنـین ویژگـی‌هـایی و همچنـین یـک معمـاری جدیـد از شـبکه‌هـای خودرمزنگار، یعنی شبکه‌های خودرمزنگار موازی، روشی به منظور تشخیص برخط صفحات جعلی اینترنتی معرفی شده است.

ادامه‌ی مقاله به صورت زیر سازمان‌دهی شده است:

در بخش دوم تعدادی از کارهای گذشته معرفی می‌شـود. در بخش سوم روش پیشنهادی شرح داده می‌شود. در بخش چهـارم نتایج گزارش شده و در بخش پنجم نتیجه‌گیری ارائه می‌شود.

۲- کارهای گذشته

ویـدر و همکـاران [۶]، طـی پژوهشـی انگیـزه‌هـای یـک مهـاجم فیشینگ در سرقت اطلاعات را در موارد زیر خلاصه نمودند:

الف) به دست آوردن نیازهای مالی[3]: در این حالـت نیازهـای مالی مهاجمین آن‌ها را بـه سـرقت از مراکـزی ماننـد بانـک‌هـا و حساب‌های مشتریان آن‌ها جلب می‌کند.

ب) پنهان مانند هویت[4]: تعدادی از مهاجمین جهت اقـدامات خرابکارانه خود نیاز به پنهان مانند هویتشان احساس می‌شـود. از این رو آن‌ها با استفاده از نام کاربری و کلمه عبور دیگـران سـعی در خریدهای اینترنتی دارند که هویت‌شان پنهان بماند.

ج) شهرت و محبوبیت[5]: تعدادی زیادی از کاربران بـه جهـت کسب شهرت و معروف شدن در انظار عمومی دست به این گونـه حملات می‌زنند.

مزیت روش ویدر و همکاران [۶]، ارایه یک معیـار مناسـب از انگیزه‌های یک مهاجم در حمـلات فیشـینگ و تعریفـی جـامع از حملات فیشینگ و دسته‌بندی آن‌ها می‌باشد با این وجود با توجه به گسترده بـودن ماهیـت ایـن حمـلات، در ایـن پـژوهش فقـط تعدادی از روش‌های فیشینگ ارایه شده است.

ویستیک و همکاران [۷]، با بررسی تعداد زیـادی از حمـلات فیشینگ یک چارچوب و سیکل برای آن‌ها تعریف نمودند. آن‌هـا نشان دادند در حملات فیشـینگ یـک چرخـه و سـیکلی جهـت حمله وجود دارد که به سیکل حملات فیشـینگ[6] موسـوم اسـت. بررسی سیکل حملات فیشینگ می‌توانـد در طراحـی روش‌هـای ضد فیشینگ[7] به کار گرفته شود. هنگامی که یک حمله فیشینگ آغاز می‌شود (مثلاً با ارسال پست الکترونیـک‌هـای فیشـینگ بـه کاربران)، اولین خط امنیتـی در مقابـل ایـن حمـلات، تشـخیص حمله فیشینگ اسـت. در تشـخیص حمـلات فیشـینگ دو روش

[1] Machine Learning
[2] Artificial Neural Network

[3] Financial gain
[4] Identity hiding
[5] Fame and notoriety
[6] life-cycle of phishing
[7] anti-phishing techniques

نرم‌افزارهای سمت کاربر[۱] و روش‌های آگاهی‌دهنده به کاربران[۲] وجود دارد.

ساهو[۳] و همکاران [۸]، روش‌های سرقت اینترنتی به کمک پست الکترونیک‌های فریبنده و تبلیغاتی را بررسی نمودند. آن‌ها برای تشخیص و شناسایی حملات فیشینگ یک دسته‌بندی جامع از انواع پست الکترونیک‌های دریافتی کاربران را ارایه نمودند. بر اساس پژوهش آن‌ها، یک پست الکترونیک از نوع فیشینگ در پنج دسته فرم اچ تی ام ال[۴]، آدرس قلابی[۵]، محتوا محور[۶]، تصویر محور[۷] و نرم افزارهای مخرب[۸] دسته‌بندی می‌شود. با وجود دسته‌بندی مناسب آن‌ها، تمایزی بین هرزنامه‌ها و پست الکترونیک‌های مبتنی بر فیشینگ در مقاله آن‌ها کم‌رنگ است.

ندا عبدالحمید و همکاران در [۹] روش‌های جستجو بر مبنای لیست‌های سیاه و سفید که قبل از این بیان شد، به عنوان راه‌حل مقابله با مشکل حملات فیشینگ درنظر گرفته شده که روش چند برچسبه طبقه‌بندی شده بر اساس طبقه‌بندی انجمنی[۹] نامیده شده است. یکی از روش‌های هوشمند بر مبنای داده‌کاوی، طبقه‌بندی انجمنی[۱۰] می‌باشد که از بازده و عملکرد خیلی خوبی برخودار بوده است. با توجه به مطالعات تجربی، طبقه‌بندی انجمنی اغلب طبقه‌بندی‌هایی شامل قوانین ساده «اگر-آنگاه» با درجه‌ی بالایی از دقت پیش بینی استخراج می‌کند. این روش قوانین جدیدی پیدا می‌کند که به بیش از یک کلاس متصل شده و به کاربر نوع جدیدی از اطلاعات را خواهد داد.

آندرانیل بوز و همکاران در [۱۰] سازمان اطلاعات متخصصان امنیتی و ضد فیشینگ پایگاه داده‌های هشدار فیشینگی تعیین کرده‌اند که هر حادثه فیشینگ گزارش شده را از نظر سطح خطر آن ارزیابی می‌کنند. از نقطه نظر افزایش تعداد حادثه‌های فیشینگ گزارش شده، نویسندگان مقاله معتقدند که چنین رویکرد ارزیابی دستی به اندازه کافی، برای ارائه گزارش‌های منظم مناسب نمی‌باشد، همچنین به اندازه کافی برای لحاظ کردن اطلاعات مالی کافی نیست. در این مقاله از یک مدل ترکیبی داده و متن‌کاوی استفاده شده که از روش استخراج عبارات کلیدی برای یافتن عبارات بامعنی طبقه‌بندی‌شده از محتوای متنی هشدارهای فیشینگ[۱۱] استفاده می‌کند و اطلاعات طبقه‌بندی شده‌ی به دست آمده را با اطلاعات مالی شرکت‌های مورد نظر ترکیب می‌کند تا با طبقه‌بندی سطح خطر این حمله و کاهش در سرمایه‌ی کارخانه که ممکن است اتفاق بیفتد به مقابله بپردازد.

استیو لاو و همکاران [۱۱] یک مدل نظری برای ارزیابی اینکه تحت تأثیر دانش مفهومی و یا رویه بر خودکارآمدی کاربران کامپیوتر برای خنثی کردن حملات فیشینگ توسعه داده شده است. این مدل بر اساس نظریه اجتناب تهدید فناوری[۱۲] نوشته شده است که رفتار فردی کاربران را برای جلوگیری از به سرقت رفتن اطلاعات‌شان شرح خواهد داد. این مدل شرح می‌دهد که چگونه هر شخص از تهدیدهای مخرب به وسیله‌ی اندازه‌گیری حفاظت[۱۳] جلوگیری خواهد کرد.

یوانچنگ لی و همکاران در [۱۲] روشی بر مبنای ماشین بردار پشتیبان برای تشخیص وب‌سایت‌های فیشینگ ارائه کرده‌اند. ماشین بردار پشتیبان یک الگوریتم طبقه‌بندی سنتی است که دارای بازدهی طبقه‌بندی بالایی است. ولی زمانی‌که در حال کار کردن بر روی مجموعه آموزش بزرگی باشد، سرعت آموزش پایین‌تری دارد. بنابراین در این مقاله بر مبنای اختلاف بین وب‌سایت‌های فیشینگ و وب‌سایت‌های اصلی از نقطه نظر ساختار پیکربندی وب، از الگوریتم ماشین بردار پشتیبان برای شناسایی و طبقه‌بندی وب‌سایت‌های فیشینگ استفاده شده است. در این الگوریتم طبقه‌بندی بردارها بر اساس خصوصیات پیکربندی صورت می‌پذیرد. این مدل نه تنها می‌تواند پیچیدگی مدل طبقه‌بندی را کاهش دهد بلکه قادر به افزایش دقت تشخیص وب‌سایت‌های فیشینگ است. در حقیقت الگوریتم ماشین بردار پشتیبان برای حل مسائل طبقه‌بندی با پیچیدگی از مرتبه‌ی ۲ یک عامل اساسی است و با توجه به مسئله‌ی افزایش زمان محاسباتی، محققان را به سوی استفاده از الگوریتمی سریع‌تر سوق داده که بتواند زمان آموزش را بدون تأثیر گذاشتن بر بازدهی طبقه‌بندی داده‌های معمولی و همچنین داده‌های ناهنجار، کاهش دهد.

کاترین پارسونز و همکاران [۱۳] تحقیقاتی را بر روی افراد انجام دادند تا در این افراد قابلیت تشخیص پست الکترونیک‌های فیشینگ و پست الکترونیک‌های واقعی را به‌وجود آورند. نتایج تحقیقات آن‌ها نشان داده است که افراد آموزش دیده بهتر توانسته‌اند تا پست الکترونیک‌های فیشینگ را از پست الکترونیک‌های واقعی تشخیص دهند. در این مقاله بر روی ۱۷۷ نفر مطالعه صورت گرفته است و تمامی این افراد در نقش دریافت‌کننده پست الکترونیک قرار گرفته‌اند با این تفاوت که نیمی از آن‌ها از امکان ارسال پست الکترونیک جعلی مطلع بوده و نیمی دیگر از این موضوع نامطلع بوده‌اند. افراد مطلع از وجود

[1] end-user client software
[2] User awareness programs
[3] Sahu
[4] HTML Form
[5] Spoofed URL
[6] Content Based
[7] Image Based
[8] Malicious Software
[9] Multi-label Classifier based Associative Classification
[10] Associative Classification (AC)
[11] Phishing risks
[12] Technology Threat Avoidance Theory (TTAT)
[13] Safeguarding Measure

پست الکترونیک‌های جعلی بازدهی بهتری در تشخیص این پست الکترونیک‌ها داشته‌اند.

زبیهی‌مایوان و دوران [۱۴]، از یک روش انتخاب ویژگی مبتنی بر نظریه‌ی مجموعه‌ی ناهنجار فازی[1] به منظور افزایش دقت در تشخیص حملات فیشینگ استفاده کرده‌اند. در این روش از سه مجموعه داده‌ی مختلف مؤثرترین ویژگی‌های موجود در تشخیص حملات فیشینگ استخراج شده است. نتایج نشان می‌دهند که این انتخاب ویژگی، نسبت به استفاده از تمام ویژگی‌ها دقت تشخیص فیشینگ را در حدود ۳٪ تا ۵٪ بهبود می‌بخشد.

لام و کتانی [۱۵] یک برنامه‌ی کاربردی تشخیص فیشینگ را توسعه داده‌اند که قابل استفاده در شبکه‌های اینترنت اشیاء است. تمرکز آن‌ها بر روی تشخیص زودهنگام فیشینگ در مرحله‌ی پست الکترونیکی است. به همین سبب تمام ویژگی‌های پست الکترونیکی از جمله عنوان، متن، فایل‌های ضمیمه شده، فرستنده، گیرنده‌های دیگر و تاریخ ارسال مورد بررسی قرار می‌گیرد و بر اساس آن‌ها وجود حمله فیشینگ گزارش می‌شود تا از ورود کاربر به صفحه‌ی جعلی جلوگیری شود.

گوپتا و جین [۱۶]، یک روش مبتنی بر موتور جستجو برای تشخیص حملات فیشینگ ارائه کرده‌اند. در این روش به کمک یک جستجوی سبک و مستقل از زبان پیوندهای مشکوک مورد ارزیابی قرار می‌گیرند به طوری که با جستجوی آن‌ها در موتور جستجوگر و استخراج ویژگی‌هایی مانند تعداد پیوندهای خارج به داخل، وجود وب‌سایت‌های مشابه از نظر محتوا و رتبه‌ی صفحات در موتور جستجوگر، جعلی بودن یا نبودن آن صفحه مشخص می‌شود. این روش توانسته است دقتی معادل با ۹۸,۱۵٪ در تشخیص صفحات جعلی به دست آورد.

یانگ و همکاران [۱۷]، یک روش مبتنی بر یادگیری عمیق و ویژگی‌های چندبعدی برای تشخیص وب‌سایت‌های فیشینگ ارائه کرده‌اند. در این روش نیازی به اجرای مرحله‌ی استخراج ویژگی نیست و شبکه عصبی عمیق اطلاعات صفحه‌ی اینترنتی را تبدیل به کد اسکی کرده و ویژگی‌های موردنیاز را از آن‌ها به صورت خودکار استخراج می‌کند. شبکه عمیق پیشنهادی در این مقاله ترکیبی از شبکه عصبی پیچشی[2] (CNN) و حافظه‌ی بلند کوتاه‌مدت[3] (LSTM) است. نتایج نشان می‌دهند که رویکرد یادگیری عمیق دقت بالاتری نسبت به روش‌های سنتی در تشخیص فیشینگ دارد.

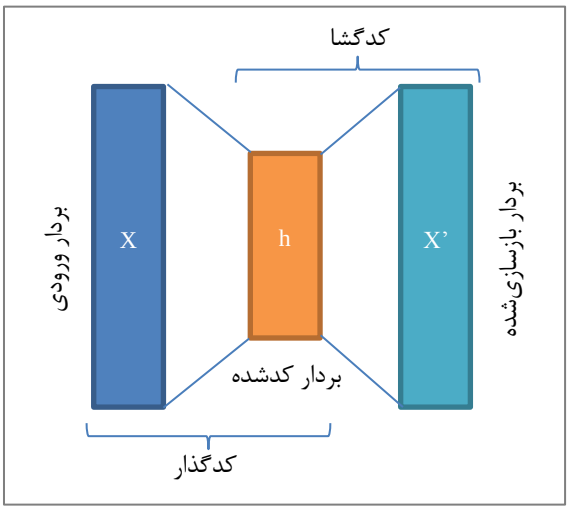
[1] Fuzzy rough set
[2] Convolutional Neural Network
[3] Long Short-Term Memory

۳- روش پیشنهادی

همان‌طور که گفته شد، روش پیشنهادی بر اساس شبکه‌های خودرمزنگار است. از این رو ابتدا این شبکه معرفی شده و پس از آن معماری موازی پیشنهادی شرح داده می‌شود.

۳-۱- شبکه‌های عصبی خودرمزنگار

خودرمزنگار نوعی شبکه عصبی است که با یک روش غیرنظارتی داده‌های ورودی را کدگذاری می‌کند. کاربردهای خودرمزنگار را می‌توان در کاهش ابعاد، استخراج ویژگی و پیش‌یادگیری خلاصه کرد. این شبکه در حالت عادی، یک بردار ورودی X را دریافت کرده و یک بردار کدشده مانند h فراهم می‌کند که معمولاً طول بردار h از X کمتر انتخاب می‌شود. بخش دوم شبکه باید بتواند بردار h را دریافت کرده و یک بردار X' که طول آن برابر با طول X است تولید کند. خطای شبکه عبارت است از اختلاف بردارهای X و X'. یعنی شبکه تلاش می‌کند وزن‌هایش به‌گونه‌ای تنظیم شود که پس کد کردن X و محاسبه h، مجدداً از h، X قابل بازیابی باشد. اگر چنین شود، می‌توان گفت که h تمام ویژگی‌های مهم X را داراست، درحالی‌که حجم آن کمتر از X است. پس می‌توان از h به عنوان یک بردار ویژگی جدید از X استفاده کرد. شکل (۱) اجزای خودرمزنگار را به خوبی نمایش می‌دهد.



شکل (۱): یک شبکه خودرمزنگار.

گاهی به منظور کاهش پارامترهای مجهول شبکه، یعنی وزن‌های کدگذار و کدگشا اگر ماتریس وزن لایه کدگذار برابر با **W** باشد، ماتریس وزنی کدگشا را برابر با $\mathbf{W}^T$ (ترآ نهاده W) در نظر می‌گیریم. با این فن، تعداد پارامترهای مجهول شبکه نصف می‌شود و در نتیجه با تعداد داده کمتری می‌توان شبکه را آموزش داد.

همان‌طور که دیده می‌شود، در این شبکه نیازی به برچسب

داده‌ها نداریم و شبکه کاملاً بدون نظارت آموزش داده می‌شود.

در ادامه روابط ریاضی این شبکه شرح داده می‌شوند:

شبکه شامل دو قسمت کدگذار و کدگشا است که به‌ترتیب آن‌ها را با E و D نمایش می‌دهیم:

$$E: \chi \rightarrow \mathcal{F} \quad (۱)$$

$$D: \mathcal{F} \rightarrow \chi \quad (۲)$$

$$D, D = \arg\min \left|\left| \boldsymbol{X} - (D \cdot E)\boldsymbol{X} \right|\right|^2 \quad (۳)$$

در ساده‌ترین حالت، بخش کدگذار یک ورودی $\boldsymbol{x} \in R^D = \chi$ را دریافت کرده و آن را به صورت $\boldsymbol{h} \in R^p = \mathcal{F}$ نگاشت می‌کند:

$$h = \sigma(Wx + b) \quad (۴)$$

به بردار به‌دست‌آمده h، معمولاً کد گفته می‌شود. σ یک تابع فعال‌سازی مانند سیگموید[1] است. W و b هم به‌ترتیب ماتریس وزنی و بردار بایاس لایه کدگذار هستند.

لایه خروجی حاصل از بخش کدگشا نیز باید رابطه زیر را ارضا کند:

$$x' = \sigma'(W'h + b') \quad (۵)$$

وزن، بایاس و تابع انتقالی لایه کدگشا می‌تواند هیچ ربطی به مقادیر لایه کدگذار نداشته باشد، اما معمولاً $W' = W^T$ در نظر گرفته می‌شود تا حجم محاسبات کاهش بیابد.

شبکه خودرمزنگار به گونه‌ای آموزش داده می‌شود که بازسازی لایه ورودی با کمترین خطا انجام شود. از این رو تابع هزینه شبکه را می‌توان به‌صورت زیر تعریف کرد:

$$L(x, x') = ||x - x'||^2 = \left|\left| x - \sigma'(W'(\sigma(Wx + b)) + b') \right|\right|^2 \quad (۶)$$

۳-۲- معماری پیشنهادی برای تشخیص فیشینگ

یکی از کاربردهای خودرمزنگار استخراج ویژگی است. به طوری که شبکه، نمونه‌های آموزشی را دریافت کرده و پس از آموزش به صورت غیرنظارتی، می‌توان از بردار رمزشده h به‌عنوان یک بردار ویژگی در یادگیری بانظارت[2] استفاده کرد.

اگر در بخش یادگیری بانظارت از یک لایه شبکه عصبی دیگر استفاده شود، یک شبکه یکپارچه به‌دست می‌آید که به آن شبکه خودرمزنگار چسبیده[3] (SAE) گفته می‌شود. شبکه‌های SAE درواقع همان شبکه‌های عصبی چندلایه پرسپترون[4] (MLP) هستند. اما مزیت آن‌ها نسبت به MLP این است که وزن‌های اولیه به‌صورت تصادفی تعیین نمی‌شود؛ بلکه به کمک یک روند پیش‌یادگیری[5] و به کمک شبکه‌های خودرمزنگار، وزن هر لایه به‌صورت مجزا تعیین می‌شود.

عملیات پیش‌یادگیری باعث می‌شود که قبل از آموزش، شبکه تا حدود زیادی با فضای داده‌های ورودی آشنا شود و درهنگام آموزش با سرعت بیشتری آموزش ببیند. از این‌رو، عملکرد بهتری نسبت به MLP خواهد داشت. با این وجود شبکه‌های SAE دارای یک عیب هستند که درواقع نوآوری این مقاله ارائه راه‌حلی برای حل این عیب است.

شبکه خود رمزنگار قادر است ویژگی‌های اساسی نمونه‌های ورودی را استخراج کند. اما وقتی کاملاً به‌صورت غیرنظارتی این کار انجام می‌شود، ممکن است برخی از مشخصه‌های اساسی که باعث تفکیک کلاس‌ها می‌شود از بین برود. از این‌رو، در این مقاله یک روش جدید استفاده از شبکه خودرمزنگار معرفی می‌شود که در آن از ساختار موازی خود رمزنگارها استفاده شده است. در این ساختار، به‌ازای هر کلاس یک شبکه خودرمزنگار وجود دارد. از آن‌جایی که مسئله این مقاله دو کلاسه است (جعلی یا معمولی)، از این‌رو، از دو شبکه خود رمزنگار موازی استفاده می‌شود. درنتیجه می‌توان گفت که معماری پیشنهادی درواقع حاوی دو شبکه SAE است که ابتدا و انتهای آن‌ها به هم چسبیده است. همچنین در هر SAE مرحله پیش‌یادگیری با داده‌های متفاوتی انجام می‌شود تا تفکیک‌پذیری بین کلاس‌ها بیشتر شده و در مرحله آموزش دقت بالاتری حاصل شود.

به‌طور مشخص برای مسئله این مقاله یعنی تشخیص صفحات جعلی اینترنتی، مراحل زیر طی می‌شود:

۱- جمع‌آوری نمونه‌های فیشینگ و آموزش یک شبکه خودرمزنگار به‌صورت غیرنظارتی (AE1): در این مرحله یک شبکه خودرمزنگار تنها با داده‌های یک کلاس (صفحات فیشینگ) آموزش می‌بینند، از این‌رو، تنها ویژگی‌های خاص صفحات فیشینگ را یاد می‌گیرد.

۲- جمع‌آوری نمونه‌های معمولی و آموزش یک شبکه خودرمزنگار دیگر به‌صورت غیرنظارتی (AE2): در این مرحله نیز مانند مرحله قبل، یک شبکه خودرمزنگار تنها با اطلاعات صفحات معمولی آموزش داده می‌شود و به این ترتیب این شبکه ویژگی‌های صفحات عادی را استخراج می‌کند.

۳- دادن تمام نمونه‌های موجود از هر دو کلاس به دو شبکه AE1 و AE2 و استخراج بردارهای h1 و h2 از آن‌ها: از آن‌جایی که کل شبکه باید برای هر دو کلاس فیشینگ و معمولی عمل کند، هر دو شبکه خود

[1] Sigmoid
[2] Supervised
[3] Stacked Auto-Encoder
[4] Multi-Layer Perceptron
[5] Pre-training

رمزنگاری که درمراحل قبل آموزش داده شدند، ویژگی‌های داده‌های ورودی از هر دو کلاس را فراهم می‌کنند تا در مراحل بعدی بر اساس این ویژگی‌ها، کلاس هر ورودی معین شود.

۴- الحاق بردارهای h1 و h2: در این مرحله یک بردار ویژگی یکپارچه از گلوگاه دو شبکه خودرمزنگار تشکیل می‌شود که به‌عنوان بردار نهایی در مرحله آموزش بانظارت و طبقه‌بندی مورد استفاده قرار گیرد.

۵- آموزش یک لایه‌ی سافت مکس[1] به‌صورت بانظارت با کل نمونه‌ها به کمک بردار الحاق شده و برچسب نمونه‌ها به‌صورت بانظارت: این مرحله خروجی شبکه پیشنهادی را تأمین می‌کند و درواقع کار طبقه‌بندی را بر اساس ویژگی‌های به‌دست آمده از شبکه‌های خودرمزنگار انجام می‌دهد.

۶- آموزش مجدد کل شبکه با کل مجموعه داده آموزش به‌صورت بانظارت برای تنظیمات نهایی وزن‌های AE1، AE2 و سافت مکس: به این مرحله تنظیم دقیق[2] گفته می‌شود و درواقع وزن‌های کل شبکه یک بار دیگر بهینه‌سازی می‌شوند. طبعاً کدگذار شبکه‌های خودرمزنگار نیز در راستای افزایش دقت کل شبکه، اندکی تغییر می‌کنند.

۷- بدیهی است که از بخش کدگشای هیچکدام از شبکه‌های خودرمزنگار در شبکه نهایی استفاده نمی‌شود. درواقع صرفاً از بخش کدگذار هر شبکه به منظور استخراج ویژگی‌های مؤثر هر کلاس استفاده شده است.

در شکل (۲) معماری پیشنهادی، نمایش داده می‌شود.

۴- نتایج

به منظور آزمایش روش پیشنهادی از مجموعه داده‌ی Phishing Websites از پایگاه UCI [۱۸] استفاده شده است.

این مجموعه داده شامل ۱۱۰۵۵ نمونه وب‌سایت است که ۶۱۵۷ نمونه از آن‌ها وب‌سایت‌های فیشینگ و ۴۸۹۸ مورد دیگر وب‌سایت‌های مشروع هستند. از هر وب‌سایت ۳۰ ویژگی استخراج شده است. این ویژگی‌ها اعداد صحیح و عمدتاً دو مقداری (صحیح و غلط[3]) هستن که به شرح زیر می‌باشند:

۴-۱- ویژگی‌های مبتنی بر نوار آدرس

- آیا از آدرس IP استفاده کرده یا خیر؟ (معمولاً دامنه‌هایی که حاوی آی‌پی هستند فیشینگ هستند).
- طول آدرس اینترنتی چقدر است؟ (معمولاً کمتر از ۵۴ کاراکتر طبیعی است، بین ۵۴ تا ۷۵ کاراکتر مشکوک و بالاتر از ۷۵ کاراکتر فیشینگ).
- آیا از ریزآدرس[4] استفاده کرده است یا خیر؟ (برخی برای این‌که طول اصلی آدرس که بزرگ است را کوچک‌تر نشان دهند از یک ریزآدرس استفاده کرده و صفحه را به صورت غیرمستقیم به آدرس اصلی هدایت می‌کنند که طول آن بزرگ است. معمولاً اگر از ریزآدرس استفاده شده باشد، سایت از نوع فیشینگ است).
- آیا از کاراکتر @ در آدرس سایت استفاده شده است؟ (معمولاً اگر استفاده شده باشد، فیشینگ است).
- آیا از کاراکتر "//" در آدرس استفاده شده است؟ (این یعنی به یک آدرس دیگر هدایت می‌شود و معمولاً اگر موقعیت این کاراکتر بعد از کاراکتر هفتم باشد، فیشینگ است).
- آیا از کاراکتر خط فاصله (-) در آدرس استفاده شده است؟ (معمولاً اگر استفاده شده باشد فیشینگ است).
- آیا از زیردامنه[5] استفاده شده است؟ (اگر یک زیردامنه استفاده شده باشد، مشکوک است. بیشتر از یک زیردامنه فیشینگ است).
- آیا از https در آدرس استفاده شده است؟ (اگر استفاده شده باشد و مدت تأیید آن بیش از ۱ سال باشد، وب‌سایت مشروع است).
- آیا مدت زمان ثبت دامنه کمتر از یک سال است؟ (معمولاً فیشرها برای مدت طولانی دامنه خریداری نمی‌کنند. پس اگر مدت اعتبار دامنه کمتر از ۱ سال باشد احتمالاً فیشینگ است).
- آیا favicon (تصویر کوچکی که کنار نوار آدرس قرار می‌گیرد) از یک آدرس دیگر ارسال شده است؟ (در این صورت احتمالاً فیشینگ است).
- آیا از درگاه غیر استاندارد استفاده شده است؟ (در این صورت احتمالاً فیشینگ است).
- آیا از عبارت https به‌عنوان بخشی از دامنه استفاده شده است؟ (در این صورت احتمالاً فیشینگ است).

۴-۲- ویژگی‌های مبتنی بر غیرطبیعی بودن صفحه

- آیا درخواست آدرس بالایی از دیگر سایت‌ها دارد؟ (معمولاً سایت‌های فیشینگ چیز زیادی از خود ندارند

[1] Softmax
[2] Fine tuning
[3] True and False
[4] TinyURL
[5] Sub-domain

و بیشتر محتوا را از سایتی که قرار است از روی آن نسخه‌ی جعلی بسازند، می‌گیرند. پس اگر درخواست آدرس زیاد باشد احتمالاً فیشینگ است).

- آیا در کد html سایت از تگ <a> استفاده شده است؟ (اگر از این تگ زیاد استفاده شده باشد به این معناست که آدرس‌های زیادی از سایت‌های دیگر درخواست می‌کند. پس احتمالاً فیشینگ است).
- آیا از تگ‌های <meta> , <script> ,<link> استفاده شده است؟ (درصورتی که زیاد استفاده شده باشد، احتمالاً فیشینگ است).
- کنترل‌کننده فرم سرور یا SFH[1] حاوی چه مقداری است؟ (اگر حاوی about: blank باشد معمولاً فیشینگ است).
- آیا از دستورات ()mail یا :mailto استفاده شده است؟ (اگر چنین باشد یعنی وب‌سایت تلاش دارد اطلاعات ورودی کاربر را به پست الکترونیکی ارسال کند. پس احتمالاً فیشینگ است).
- آیا از iFrame استفاده شده است؟ (معمولاً فیشینگ‌ها حاوی iframe هستند).

۴-۳- ویژگی‌های مبتنی بر دامنه

- عمر دامنه چقدر است؟ (به کمک پایگاه داده WHOIS می‌توان عمر یک دامنه را مشاهده کرد. اگر عمر این دامنه بزرگتر از ۶ سال باشد، می‌توان گفت وب‌سایت مشروع است).
- آیا اطلاعاتی برای DNS[2] وجود دارد؟ (بر اساس WHOIS اگر دامنه متعلق به یک شخص مشخص باشد، می‌توان آن را مشروع دانست اما اگر صاحب دامنه یا ارائه‌دهنده‌ی آن نامعلوم یا ثبت نشده باشد، فیشینگ است).
- رتبه‌ی الکسا چقدر است؟ (اگر ترافیک یا بازدید سایت بالا باشد معمولاً سایت مشروع است. بر اساس رتبه‌ی الکسا معمولاً وب‌سایت‌هایی که رتبه بالاتر از ۱۰۰۰۰۰ دارند مشکوک به فیشینگ هستند).
- مقدار رتبه‌ی صفحه[3] چقدر است؟ (اگر کمتر از ۰/۲ باشد معمولاً فیشینگ است).
- آیا در گوگل فهرست شده است؟ (معمولاً فیشینگ‌ها به خاطر عمر کوتاه‌شان در گوگل فهرست نمی‌شوند).
- تعداد پیوندهای خارج به داخل چقدر است؟ (اگر سایت‌های دیگری وجود داشته باشند که به سایت موجود پیوند داده باشند احتمالاً وب‌سایت موجود مشروع است).
- اطلاعات آماری در مورد وب‌سایت وجود دارد یا خیر؟ (برخی سایت‌ها از قبل توسط دیگران به عنوان فیشینگ معرفی شده‌اند که می‌تواند اطلاعات مفیدی باشد).

همانطور که دیده می‌شود، مجموعه داده‌ی مورد استفاده اکثر اطلاعات مرتبط با حملات فیشینگ را در قالب ۳۰ ویژگی استخراج کرده است. با در نظر گرفتن هر ۳۰ ویژگی می‌توان مدلی با دقت بالا ایجاد کرد که بتواند وب‌سایت‌های فیشینگ را به صورت برخط تشخیص دهد.

روش پیشنهادی به کمک نرم‌افزار MATLAB پیاده‌سازی شده است و دیگر تنظیمات به شرح زیر است:

- اندازه‌ی لایه پنهان در هر دو شبکه خودرمزنگار برابر با ۲۰ در نظر گرفته شده است. یعنی بردار ورودی که ۳۰ ویژگی دارد، تبدیل به یک بردار ویژگی با طول ۲۰ خواهد شد. با الحاق دو شبکه، بردار ویژگی نهایی دارای ۴۰ ویژگی است.
- تابع انتقالی لایه‌های خودرمزنگار از نوع سیگموید است. از این رو مقادیر ویژگی‌های بردارهای ۲۰ تایی بین ۰ تا ۱ خواهد بود.

ارزیابی نتایج نیز به کمک الگوریتم اعتبارسنجی متقابل ۱۰ دوره[4] انجام شده است و نتایج به‌دست آمده میانگینی از تمام ۱۰ دور ارزیابی است.

همچنین از مقادیر صحت متوسط[5]، دقت[6] و فراخوانی[7] استفاده شده که به‌صورت زیر محاسبه می‌شوند:

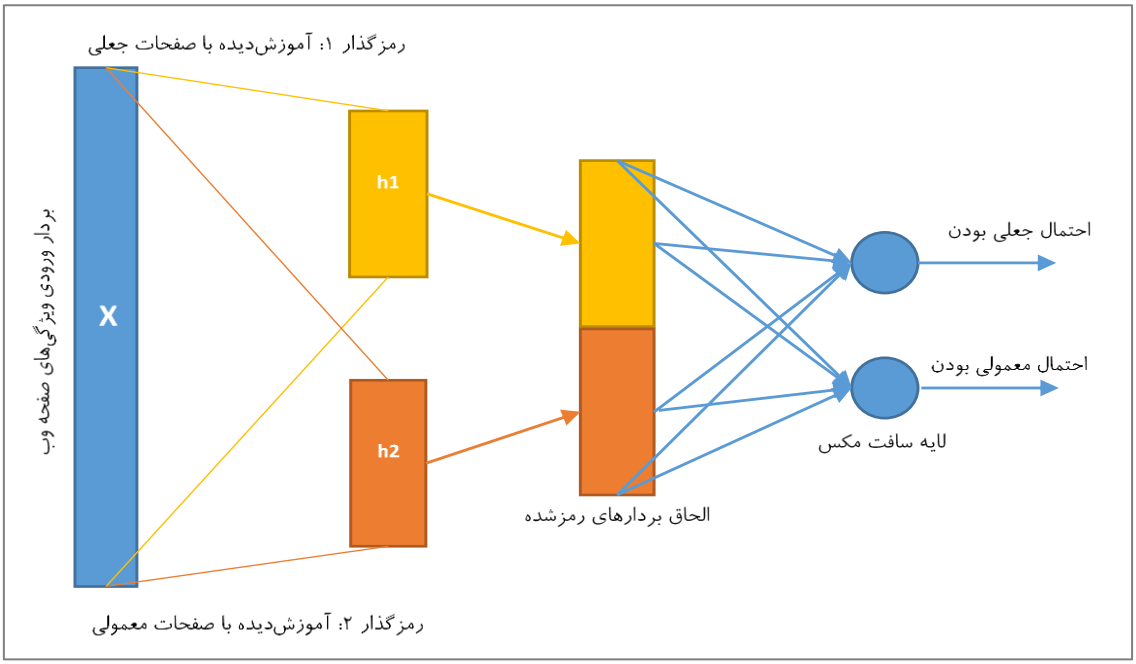
$$Accuracy= (TP+TN)/ (TP+TN+FP+FN) \quad (۷)$$

$$Precision= TP/ (TP+FN) \quad (۸)$$

$$Recall= TP/ (TP+FP) \quad (۹)$$

[1] Server Form Handler
[2] Domain Name Service
[3] PageRank

[4] 10-fold cross-validation
[5] Accuracy
[6] Precision
[7] Recall



شکل (۲): معماری پیشنهادی برای تشخیص صفحات جعلی.

که:

TP: نمونه‌های جعلی است که به درستی جعلی تشخیص داده شده است.

TN: نمونه‌های معمولی است که به درستی معمولی تشخیص داده شده است.

FP: نمونه‌های جعلی است که به اشتباه معمولی تشخیص داده شده است.

FN: نمونه‌های معمولی است که به اشتباه جعلی تشخیص داده شده است.

در شکل (۳) ماتریس تداخل برای نتایج روش پیشنهادی، نمایش داده می‌شود.

همان‌طور که دیده می‌شود، روش پیشنهادی توانسته در مجموع ٪۹۶/۵ صحت متوسط را فراهم کند. از بین ۶۱۵۷ نمونه فیشینگ ۶۰۰۷ نمونه (تقریباً ٪۹۷/۶) به‌درستی تشخیص داده شده‌اند.

یعنی تنها ۱۴۹ نمونه به اشتباه مشروع در نظر گرفته شده‌اند. نمونه‌های مشروع نیز با ٪۹۵ دقت تشخیص داده شده‌اند.

برچسب خروجی			
معمولی (۰)	۴۶۵۱	۱۴۹	٪۹۶/۹
جعلی (۱)	۲۴۳	۶۰۰۷	٪۹۶/۱
	٪۹۵/۰	٪۹۷/۶	**٪۹۶/۵**
	معمولی (۰)	جعلی (۱)	
	برچسب واقعی		

شکل (۳): ماتریس تداخل به‌دست‌آمده از اجرای روش پیشنهادی.

جدول (۱) نتایج به‌دست‌آمده از روش پیشنهادی را با روش‌های متداول دیگر که برای حل مسئله تشخیص صفحات جعلی استفاده شده، مقایسه کرده است. این روش‌ها عبارتند از کا- نزدیکترین همسایه (KNN)، شبکه عصبی چندلایه (MLP)، درخت تصمیم (DT) و ماشین بردار پشتیبان (SVM) .

همان‌طور که دیده می‌شود روش پیشنهادی بالاترین صحت و دقت را در بین الگوریتم‌های طبقه‌بندی دارد. علت این است که به کمک روش پیشنهادی تفکیک‌پذیری بین کلاس‌های فیشینگ و معمولی افزایش یافته است و برای یادگیری ویژگی‌های هرکدام از این کلاس‌ها یک شبکه خود رمزنگار مجزا در نظر گرفته شده است.

در روش‌های سنتی مانند ماشین‌بردار پشتیبان، ویژگی‌های اولیه به‌همان شکل مورد استفاده قرار می‌گیرند و اگر بخشی از کلاس‌ها روی هم‌رفتگی داشته باشند، الگوریتم ناچار است بخشی از داده‌ها را درون مرز کلاس‌ها بپذیرد و در نهایت منجر به ایجاد خطا در ماشین‌بردار پشتیبان می‌شود. اما در روش پیشنهادی با ایجاد تفکیک بیشتر بین کلاس‌ها، این مسئله تا حدودی برطرف می‌شود.

در نتایج دیده می‌شود که شبکه عصبی MLP عملکرد بسیار ضعیفی داشته و در واقع بیشتر صفحات را معمولی تشخیص داده است. به‌همین دلیل فراخوانی آن بسیار بزرگ شده است، اما دقت آن در تشخیص صفحات فیشینگ که هدف اصلی مقاله است، بسیار کوچک است. این درحالی است که روش پیشنهادی نیز نوعی شبکه عصبی است. تفاوت روش پیشنهادی با MLP اولاً در روند پیش‌یادگیری است که باعث می‌شود وزن‌های اولیه شبکه عصبی، دیگر تصادفی نباشند و ثانیاً ایجاد ساختار موازی در شبکه، برای افزایش تفکیک‌پذیری کلاس‌ها می‌باشد. در MLP وزن‌های اولیه تصادفی انتخاب می‌شوند و این باعث می‌شود شبکه دچار مشکلاتی از قبیل محوشدگی گرادیان[1] یا بیش‌برازش[2] شود. یعنی گرادیان آن‌قدر کوچک شود که لایه‌های اولیه به خوبی به‌روزرسانی نشوند یا این‌که شبکه آن‌قدر بر روی داده‌های آموزش تنظیم شود که کوچک‌ترین تغییری در داده‌های آزمون باعث ایجاد خطا فاحش در تشخیص شود. درواقع رویکرد پیش‌یادگیری که در این مقاله مورد استفاده قرار گرفت، یکی از راه‌حل‌هایی است که برای حل مشکلات محوشدگی گرادیان و بیش‌برازش استفاده می‌شود و نتایج به خوبی تأثیر این رویکرد را نشان می‌دهند.

الگوریتم‌های KNN و DT به سبب سادگی مجموعه داده مورد استفاده عملکرد بسیار خوبی داشته‌اند. زیرا ویژگی‌های مجموعه داده عمدتاً دارای مقادیر ۱ و ۱- هستند که این مقادیر نیز با فیشینگ بودن یا نبودن صفحه ارتباط مستقیم دارند. از این‌رو یافتن قوانین شرطی اگر- آنگاه برای طبقه‌بندی صفحات اینترنتی به سادگی قابل انجام است. به همین سبب است که DT که در واقع مبتنی بر شرط‌های اگر-آنگاه است توانسته عملکرد خوبی در تشخیص فیشینگ داشته باشد. طبعاً در مجموعه داده‌های پیچیده‌تر DT به این خوبی عمل نخواهد کرد. همچنین عملکرد خوب KNN نشان از این است که داده‌های موجود در مجموعه داده شباهت بسیار زیادی با یکدیگر داشته‌اند. درواقع به‌ازای هر داده آزمون، یک داده آموزش مشابه وجود داشته است. به‌همین سبب الگوریتم دقت بالایی گزارش داده است. اما اگر شگردهای جدیدی در فیشینگ اعمال شود که نمونه آن در داده‌های آموزش وجود نداشته باشد، احتمالاً KNN نیز دچار خطا بیشتری می‌شود.

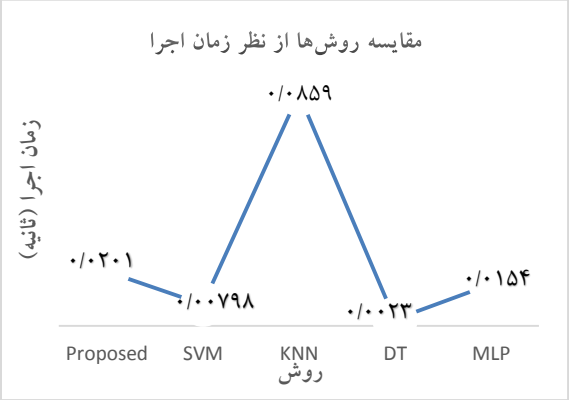
با توجه به اینکه در روش پیشنهادی، چند بار عملیات آموزش انجام می‌شود زمان اجرای بالایی در مرحله آموزش دارد. اما در زمان آزمون و استفاده از مدل، همانند سایر روش‌ها به‌اندازه کافی سریع است. نمودار شکل (۴)، سرعت اجرای روش‌های مختلف را در زمان استفاده (آزمون) برای طبقه‌بندی ۱۱۰۵ نمونه را نمایش می‌دهد.

همان‌طور که دیده می‌شود یکی از اصلی‌ترین رقیب‌های روش پیشنهادی از نظر دقت (یعنی KNN) بالاترین زمان اجرا را در زمان استفاده دارد و باعث می‌شود که در استفاده‌های برخط مورداستفاده نباشد. اما روش پیشنهادی علاوه‌بر اینکه دقت بالایی در تشخیص صفحات جعلی دارد، همانند سایر روش‌های سریع زمان اجرای مناسبی دارد و قادر است به‌صورت بلادرنگ مورد استفاده قرار گیرد.

جدول (۱): مقایسه روش‌ها.

الگوریتم	صحت	دقت	فراخوانی	TP	TN	FP	FN
روش پیشنهادی	**۹۶/۵**	**۹۷/۶**	۹۶/۱	**۶۰۰۷**	۴۶۵۱	**۱۴۹**	۲۴۳
SVM	۹۲/۷	۹۴/۷	۹۲/۴	۵۸۲۸	۴۴۱۵	۳۲۸	۴۷۹
KNN	۹۶/۰	۹۶/۹	۹۵/۹	۵۹۶۶	۴۶۴۰	۱۹۰	۲۵۴
DT	۹۵/۵	۹۶/۸	۹۵/۲	۵۹۶۰	۴۵۹۰	۱۹۶	۳۰۴
MLP	۶۲/۷	۳۳/۳	**۹۹/۵**	۲۰۴۹	**۴۸۸۴**	۴۱۰۷	**۱۰**

[1] Gradient Vanishing
[2] Overfitting



شکل (۴): نمودار مقایسه زمان اجرای الگوریتم‌ها در زمان آزمون.

۵- نتیجه‌گیری

همان‌طور که دیده می‌شود، روش پیشنهادی نسبت به سایر روش‌های مرسوم، از جمله ماشین بردار پشتیبان قوی‌تر عمل کرده است. تحلیل‌های انجام شده بر روی زمان اجرا نیز به خوبی نشان می‌دهد که روش پیشنهادی پس از طی کردن مرحله آموزش قادر است به‌صورت بلادرنگ و برخط، همانند سایر روش‌ها مورد استفاده قرار گیرد. علت عملکرد مؤثر روش پیشنهادی این است که شبکه‌های اتوانکدر در ابتدای کار ویژگی‌های مهم هر کلاس را استخراج می‌کنند و این باعث می‌شود که تفکیک‌پذیری کلاس‌ها در ابتدای آموزش شبکه عصبی بالاتر برود. از این‌رو طبقه‌بندی، در نهایت با دقت بالاتری انجام می‌شود.

۶- مراجع

[1] H. Bahrami, "Improving IWO Algorithm in Generation Optimization Problems using Experiment Design," M.Sc. Thesis, Ferdosi Univ., Mashhad, 2010. (In Persian)

[2] M. Moghimi, H. Albaripoor, and M. R. Amin-Naseri, "Expert System Designing for Phishing Attacks Detection in E-Banking," Journal of Iranian Association of Electrical and Electronics Engineers, vol. 12(2), 2015. (In Persian)

[3] B. Bahreini, A. Malahzadeh, and M. Soleimani, "Antenna Array Designing using IWO algorithm for MIMO systems in 5.8 GHz frequency," 18th Iran Electrical Engineering Conference, 2010. (In Persian)

[4] M. Alsharnouby, F. Alaca, and S. Chiasson, "Why phishing still works: user strategies for combating phishing attacks," Int. J. Hum-Comput. St., vol. 82, pp. 69-82, 2008.

[5] H. Rahimi, I. Rahmi, and J. H. Abawajy, "An approach for profiling phishing activities," Cumpot. Secur., vol. 45, pp. 27-41, 2014.

[6] W. D. Yu, S. Nargundkar, and N. Tiruthani, "A phishing vulnerability analysis of web based systems," Cumput. Commun., pp. 326-331, 2008.

[7] B. Vucetic and J. Yuan, "Space-time coding," JWS, 2003.

[8] K. R. Sahu and J. Dubey, "A Survey on Phishing Attacks," Int. J. Comut. Appl., vol. 88, 2014.

[9] N. Abdelhamid, A. Aladdin, and T. Fadi, "Phishing detection based associative classification data mining," Expert. Sys. Appl., vol. 41 pp. 5948-5959, 2014.

[10] C. L. Tan, C. Kang Leng, and W. KokSheik, "PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder," Decis. Support Syst., vol. 88, pp.18-27, 2016.

[11] N. A. G. Arachchilage and S. Love, "Security awareness of computer users: A phishing threat avoidance perspective," Comput. Human Behav., vol. 38, pp. 304-312, 2014.

[12] Y. Li, L. Yang, and J. Ding, "A minimum enclosing ball-based support vector machine approach for detection of phishing websites," OPTIC (Stuttg), vol. 127.1, pp. 345-351, 2016.

[13] K. Parsons, A. McCormac, M. Pattinson, M. Butavicius, and C. Jerram, "The design of phishing studies: Challenges for researchers," Comput. Secur., vol. 52, pp.194-206, 2016.

[14] M. Zabihimayvan and D. Doran, "Fuzzy rough set feature selection to enhance phishing attack detection," Fuzz-IEEE, pp. 1-6, 2019.

[15] T. Lam and H. Kettani, "PhAttApp: A Phishing Attack Detection Application," 3rd International Conference on Information System and Data Mining, pp. 154-158, 2019.

[16] B. Gupta and A. K. Jain, "Phishing Attack Detection using a Search Engine and Heuristics-based Technique," J. Inf. Technol-Uk, vol. 13, pp. 94-109, 2020.

[17] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," IEEE ACCESS, vol. 7, pp. 15196-15209, 2019.

[18] UCI Machine Learning Repository, "Phishing Websites Dataset," 2015. archive.ics.uci.edu/ml/datasets/phishing+websites.

Journal of "Innovations of Applied Information and Communication Technologies"
Vol. 1, No. 2, 2019 (Serial No. 2)

Fake Webpages Detection Using Parallel Autoencoder Neural Networks

N. Amiri*, I. Naderi
Islamic Azad University, Borojerd Branch

Abstract

Fake web pages attempt to steal a user's confidential information such as bank account password and email password. These fake pages are actually made similar to the pages of reputable websites such as online payment portals, Yahoo and Google, and in such a way users are drawn to these pages. This type of Internet attack is called phishing attacks. Online detection of fake pages with the help of a smart software can prevent the theft of user information and increase security in the web space. In this paper, a new approach based on autoencoder neural networks is introduced. The proposed method employs two Parallel Autoencoder (PAE) networks, one of which is trained with regular pages and the other with fake pages. At the time of detection, the type of input web page is determined based on the encoded vectors obtained from both AEs in the parallel network and a layer of conventional artificial neural network such as Softmax. In practical applications, whenever such a fake page is detected, it is promptly warned or blocked through the browser. Experimental results of the proposed method with the help of "Phishing Websites" dataset and accuracy, precision and recall criteria show that PAE networks perform better than other machine learning methods in detecting fake web pages.

Keywords: Fake Webpage, Phishing Attack, Machine Learning, Parallel Autoencoder Neural Networks

Corresponding author E-mail: amirinoshin@iaud.ac.ir