شناسایی و تشخیص وبسایت های مخرب با استفاده از پایگاه دانش و یادگیری ماشین و عمیق

روزبه توکلی ، دکتر حمید طباطبایی

rouzbehtava@gmail.com

۱_ دانشجوی کارشناسی ارشد مهندسی نرم افزار ، دانشگاه ازاد اسلامی واحد مشهد ، ایران

۲_ دکتر حمید طباطبایی معاونت آموزشی گروه مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد قوچان ، ایران

چکیده :

مهاجمان فیشینگ لینک های فیشینگ را از طریق ایمیل، پیام های متنی و پلتفرم های رسانه های اجتماعی پخش می کنند. آنها از مهارت های مهندسی اجتماعی برای فریب کاربران برای بازدید از وب سایت های فیشینگ و وارد کردن اطلاعات شخصی حیاتی استفاده می کنند. در پایان، اطلاعات شخصی دزدیده شده برای کلاهبرداری از اعتماد وب سایت های معمولی یا مؤسسات مالی برای به دست آوردن مزایای غیرقانونی استفاده می شود. با توسعه و کاربردهای فناوری یادگیری ماشین، بسیاری از راه حل های مبتنی بر یادگیری ماشین برای تشخیص فیشینگ ارائه شده است. برخی از راه حل ها بر اساس ویژگی های استخراج شده توسط قوانین هستند و برخی از ویژگی ها نیاز به تکیه بر خدمات شخص ثالث دارند که باعث بی ثباتی و مشکلات وقت گیر در سرویس پیش بینی می شود. در این مقاله، یک چارچوب مبتنی بر یادگیری عمیق برای شناسایی وب‌سایت‌های فیشینگ پیشنهاد کرده است. همچنین این چارچوب را به عنوان یک افزونه مرورگر پیاده‌سازی کرده است که قادر به تعیین خطر فیشینگ در زمان واقعی هنگامی که کاربر از یک صفحه وب بازدید می‌کند و یک پیام هشدار می‌دهد، وجود دارد یا خیر. سرویس پیش‌بینی بی‌درنگ استراتژی‌های متعددی را برای بهبود دقت، کاهش نرخ هشدار نادرست، و کاهش زمان محاسبه، از جمله فیلتر کردن لیست سفید، رهگیری لیست سیاه و پیش‌بینی یادگیری ماشینی MLترکیب می‌کند. در ماژول پیش‌بینی ML، ما چندین مدل یادگیری ماشین را با استفاده از چندین مجموعه داده مقایسه کردیم. از نتایج تجربی، مدل RNN-GRU بالاترین دقت ۹۹,۱۸٪ را به دست آورد که امکان سنجی راه حل پیشنهادی را نشان می دهد. همچنین در مدل دیگری از پایگاه دانش استفاده شد با چارچوب جدید بنام DOCRIW معرفی شد، DOCIRW شامل یک پایگاه داده دانش است که با استفاده از اطلاعات جمع‌آوری‌شده از وب‌سایت‌هایی که محتوای غیرقانونی و مخرب ارائه می‌کنند، ساخته شده است ، و همچنین یک مدل یادگیری ماشین هم برای آن انجام و بهترین نتیجه برای آن در این مقاله گزارش شده است .

کلمات کلیدی :

پایگاه دانش ، شناسایی ، تشخیص ، وبسایت های مخرب ، الگوریتم شناسایی ، یادگیری ماشین ، یادگیری عمیق

۱_ مقدمه :

خدمات اینترنتی تغییرات شگرفی را در زندگی مردم به ارمغان آورده است. اکثر سرویس های آنلاین، کاربران را از طریق سیستم عضویت مدیریت می کنند و کاربران فردی برای دریافت این خدمات شخصی سازی شده باید ثبت نام کرده و وارد شوند. بنابراین، افراد باید هنگام بهره مندی از این خدمات راحت و کارآمد، اطلاعات شخصی خود را ارائه دهند. در یک محیط شبکه ایمن، انتقال و ذخیره سازی اطلاعات توسط فناوری امنیت شبکه محافظت می شود. با این حال، مجرمان سایبری زیادی وجود دارند که از روش های مختلفی برای حمله و سرقت اطلاعات شخصی استفاده می کنند. فیشینگ یکی از روش های حمله سایبری است که یک وب سایت معمولی را شبیه سازی می کند تا کاربران را برای ارائه اطلاعات شخصی فریب دهد. از زمان ظهور حملات فیشینگ بیش از ده سال پیش، کارشناسان امنیت شبکه از روش‌های فنی برای رهگیری حملات استفاده می‌کنند. فناوری حمله و فناوری ضد حمله دائما در حال تغییر و بهبود هستند. متأسفانه هیچ فناوری مؤثری وجود ندارد که بتواند به طور کامل از حملات فیشینگ جلوگیری کند. از گزارش‌های امنیتی شبکه در سال‌های اخیر، می‌توان دریافت که ضررهای اقتصادی ناشی از حملات فیشینگ بسیار زیاد است (S. Marchal, ۲۰۱٤). طبق گزارش فعالیت فیشینگ از APWG، هر ماه بیش از ۱۰۰۰۰۰ پیوند فیشینگ وجود دارد و در سال گذشته روند رو به رشدی داشته است. گـــــزارش سالانه ۲۰۲۰ مرکز شکایت جرایم اینترنتی نشان داد که ضرر اقتصادی ناشی از حملات فیشینگ بیش از ۵۴ میلیون دلار بوده است. ابزارهای رایج برای انتشار پیوندهای فیشینگ ایمیل، پیام های متنی و پلتفرم های رسانه های اجتماعی هستند. محتوای نسخه ویرایش شده توسط مهاجم از طریق مهندسی اجتماعی به این معنی است که کاربر بسیار مشتاق است که پس از دریافت اطلاعات روی پیوند فیشینگ کلیک کند. بنابراین، هنگامی که لینک فیشینگ در مرورگر قابل دسترسی است، تشخیص خطر از طریق فناوری امنیت شبکه و هشدار به کاربر یک فناوری ضد حمله بسیار موثر برای جلوگیری از افشای اطلاعات شخصی کاربر است. هسته اصلی روش های سنتی تشخیص URL های فیشینگ بر اساس قوانین است. تولید این قوانین را می توان به عنوان تجزیه ویژگی ها از URL و کد منبع صفحه وب و مقایسه ارزش ویژگی با یک آستانه تجربی خلاصه کرد. از گزارش تحقیقات دانشگاهی می توان دریافت که تعداد قوانین مؤثر در حدود ۱۰۰ است (Safara, ۲۰۲۱) - (Mohammad, ۲۰۱٤). مجرمان سایبری همچنین می توانند استراتژی های حمله جدیدی را بر اساس این قوانین توسعه دهند. قوانین قابل تفسیر هستند و منطق قوانین محدود است، بنابراین روش های تشخیص مبتنی

بر قوانین می توانند به راحتی کرک شده و توسط مهاجمان اســـتفاده شوند. به عـــنوان مثال، ویـــژگی طرحواره URL، HTTPS است که در بسیاری از مقالات تحقیقاتی مورد اســـتفاده قرار می گیرد و از اهمیت بالایی برخوردار است. با این حال، گزارش APWG نـــشان می دهد که به طور متوسط ۸۳ درصد از وب سایت های فیشینگ در سه ماهه

اول سال ۲۰۲۱ از طرح HTTPS استفاده کرده اند. با توسعه سریع یادگیری ماشینی، کاربردهای بیشتری در زمینه امنیت سایبری وجود دارد. برخی از محققان و کارشناسان راه‌حل‌هایی را برای شناسایی لینک‌های فیشینگ بر اساس یادگیری ماشین پیشنهاد کرده‌اند و بسیاری از مقالات مجلات دانشگاهی نشان می‌دهند که راه‌حل‌های مبتنی بر یادگیری ماشینی به دقت بالایی دست یافته‌اند (B. B. Gupta, ۲۰۲۱). با این حال، در سناریوی کاربردی یک محیط بلادرنگ، هنوز چالش های زیادی وجود دارد. برای مثال، سیستم بلادرنگ نیاز دارد که زمان پاســـخگویی ســـرویس پیش‌بینی‌کننده در حد میلی‌ثانیه باشد . نرخ مـــثبت کـــاذب بالا بر تـجربه کاربر و اعـــتماد کاربر تـــأثیر می گذارد. در این مقاله، در اینجا یک چارچوب مبتنی بر یادگیری عمیق را برای شناسایی پیوندهای فیشینگ در یک محیط وبگردی بلادرنگ پیشنهاد می‌کند. ما یک افزونه مرورگر برای دریافت اطلاعات مشتری، تماس با سرویس پیش‌بینی پس‌زمینه و نمایش نتایج پیش‌بینی به کاربران ایجاد کردیم. هنگامی که نشانی اینترنتی برگه فعلی مرورگر پیش‌بینی می‌شود که یک پیوند فیشینگ باشد، صفحه فعلی یک اخطار آشکار دریافت می‌کند. نتیجه پیش‌بینی توسط سرویس پیش‌بینی اصلی که یک مدل یادگیری ماشین آموزش دیده را فراخوانی می‌کند، به دست می‌آید. ما چندین مدل را با مجموعه داده های متعدد برای مقایسه و پشتیبان معرفی کردیم. از نتایج تجربی این نتیجه حاصل می شود که مدل RNN-GRU بالاترین میزان دقت ۹۹,۱۸ را به دست می آورد که بهتر از SVM، رگرسیون لجستیک، جـــنگل تـــصادفی است. مـــشارکت های این مـــقاله عـــبارتند از : الف : یک چارچوب مبتنی بر یادگیری عمیق برای شناسایی URL های فیشینگ. ما مدل ها را با استفاده از هفت مجموعه داده سفارشی تولید شده از چهار منبع داده موجود، آموزش و آزمایش کردیم و با مدل RNN-GRU به بـــالاترین دقت ۹۹,۱۸ درصد دست یافتیم. ب : اجرای نمونه اولـیه چـارچوب پیشنهادی به عـــــنوان افزونه مـــرورگر کـــروم. کارهای انجام شده : بخش دوم کارهای مرتبط با تمرکز بر مدل های یادگیری عمیق و چارچوب های زمان واقعی را خـــلاصه می کنیم. بخش سوم طراحی و معماری چارچوب پیشنهادی را ارائه می‌کند. بخش چهارم اجرای نمونه اولیه را مورد بحث قرار می دهد، از جمله برخی چارچوب ها، سرویس ها و ابزارهای منبع باز که ما استفاده کرده است. نتایج تجربی و تجزیه و

تحلیل در بخش پنجم گزارش شده است. همچنین در بخش بعد مدل DOCIRW را بررسی می کند در نهایت، بخش ششم مقاله را به پایان می‌رسانیم و ایده‌هایی برای کار آینده ارائه می‌دهیم .

۲_ کارهای مرتبط :

حملات فیشینگ یک مشکل جدی است و فناوری شناسایی و رهگیری حملات فیشینگ به طور مداوم وجود دارد. این دقیق ترین و سریع ترین راه برای فیلتر کردن URL های خوب از طریق لیست سفید و مسدود کردن URL های فیشینگ از طریق لیست سیاه است. با این حال، روش لیست نمی‌تواند پیوندهای فیشینگ جدید را شناسایی کند، و به دلیل هزینه کم ایجاد یک URL فیشینگ، مهاجم به استفاده از یک پیوند فیشینگ چندین بار متکی نیست. گزارش های تحقیقاتی زیادی بر اساس یادگیری ماشین منتشر شده است و نتایج با دقت بالایی در آزمایش ها به دست آمده است. با این حال، در محیط واقعی شبکه، هنوز هم هر ساله قربانیان بسیاری از حملات فیشینگ وجود دارد که باعث خسارات اقتصادی می شود. هنوز شکاف مشخصی بین نتایج داده های تجربی و راه حل های واقعی امنیت شبکه وجود دارد. بنابراین، مطالعه راه حل های ضد فیشینگ در یک محیط بلادرنگ بسیار مهم است .

الف) روش های مبتنی بر یادگیری ماشین :

در این بخش، برخی از پیشرفته ترین راه حل های مبتنی بر یادگیری عمیق را برای تشخیص وب سایت های فیشینگ بررسی کرده است. Bu و Cho (Cho, ۲۰۲۱) یک مدل رمزگذار خودکار عمیق را برای شناسایی حملات فیشینگ روز صفر پیشنهاد کردند و دقت ۹۷,۳۴٪ را به دست آوردند. آنها ویژگی‌های سطح کاراکتر را از رشته‌های URL استخراج کردند و آزمایش‌هایی را روی سه مجموعه داده مختلف جمع‌آوری‌شده از Storm Phish (S. Marchal, ۲۰۱٤)، ISCX-URL-۲۰۱٦ (URL | ۲۰۱٦ | Dataset | Research)، و Tank Phish (Oct. ۲۰, ۲۰۲۱) انجام دادند. آنها از تحلیل منحنی مشخصه عامل گیرنده و اعتبارسنجی متقاطع N برابر برای ارزیابی نتایج تجربی استفاده کردند. با مقایسه ریشه میانگین مربعات خطا (RMSE) در مرحله بازسازی بین URL های قانونی و URL های فیشینگ، آنها دریافتند که RMSE به طور قابل توجهی برای URL فیشینگ افزایش یافته است.

سامشا و همکاران مدل های یادگیری عمیق را برای شناسایی وب سایت های فیشینگ تنها با استفاده از ده ویژگی استخراج شده از HTML و یک سرویس شخص ثالث معرفی کرد. آنها سه مدل یادگیری عمیق را مقایسه کردند و وزن ۱۸ ویژگی

را محاسبه کردند. نتایج تجربی نشان داد که مدل حافظه کوتاه مدت بلند (LSTM) بالاترین دقت ۹۹,۵۷٪ را به دست آورد. با این حال، آنها فقط از یک مجموعه داده منتشر شده با ۳۵۲۶ نمونه استفاده کردند. بدیهی است که مجموعه داده برای آموزش یادگیری عمیق بسیار کوچک است. میزان دقت بالا در نتایج تجربی ممکن است به دلیل توزیع ناهموار و تنوع ضعیف داده های آزمایش باشد.

آدبووال و همکاران الگوریتم شبکه عصبی کانولوشنال (CNN) و حافظه کوتاه مدت (LSTM) را برای طبقه بندی وب سایت های فیشینگ ترکیب کرد. طبقه‌بندی‌کننده ترکیبی با استفاده از ویژگی‌های تصویر، فریم و متن، دقت ۹۳,۲۸ درصد و میانگین زمان محاسباتی ۲۵ ثانیه را به دست آورد. آنها URL ها را از Tank Phish و Crawl Common جمع آوری کردند و ویژگی های تصویر را از URL ها استخراج کردند. ویژگی های تصویر برای تغذیه مدل CNN آفلاین استفاده می شود و ویژگی های متنی به طبقه بندی کننده LSTM کمک می کند. نکته ابتکاری این راه حل، ترکیب ویژگی های عکس و متن است. با این حال، از نتایج تجربی، هنوز جا برای بهبود در میزان دقت وجود دارد زمان محاسبه برای برآورده کردن الزامات محصولات پیش‌بینی بلادرنگ بسیار طولانی است.

ب) چارچوب و سیستم ها :

هنگام تشخیص اینکه آیا یک صفحه وب در معرض خطر حملات فیشینگ قرار دارد یا خیر، سرویس اصلی یک سرویس پیش بینی مبتنی بر یادگیری ماشین است. زمان پاسخگویی سرویس پیشگو مهمترین شاخص برای سنجش امکان سنجی این سیستم بلادرنگ است. آتیموراتانا و همکاران (D. N. Atimorathanna, ۲۰۲۰) یک سیستم حفاظتی ضد فیشینگ را معرفی کرد که شامل یک افزونه مرورگر وب، یک افزونه شناسایی ایمیل، فیلترها و یک سرور تشخیص فیشینگ مبتنی بر یادگیری ماشین است. افزونه مرورگر برای استخراج URL فعلی، گرفتن اسکرین شات و ذخیره سابقه بازدید کاربر به عنوان نمایه در سمت سرویس گیرنده استفاده می شود. سرور عمدتاً از فرآیندهای زیر برای شناسایی پیوندهای فیشینگ استفاده می کند: (۱) استفاده از لیست سیاه و لیست سفید خدمات شخص ثالث برای فیلتر کردن URL های جدید. (۲) استفاده از یک مدل یادگیری ماشین بر اساس ۱۳ ویژگی برای پیش بینی اینکه آیا URL یک پیوند فیشینگ است یا خیر. (۳) استفاده از فناوری بینایی رایانه برای تشخیص آرم وب سایت و مقایسه شباهت اسکرین شات از صفحات وب. آشکارساز لوگو در مقاله برای شناسایی ۲۰ بانک آنلاین معروف و برخی از آرم های رایج وب سایت استفاده می شود. نویسندگان پایگاه داده خود را برای آموزش مدل تشخیص لوگو جمع آوری و ایجاد کردند و میزان دقت بیش از ۹۵٪ را به دست آوردند. مقایسه

شباهت دو اسکرین شات از کتابخانه OpenCV پایتون استفاده می کند. نتایج تجربی تحلیلگر URL نشان داد که طبقه‌بندی‌کننده Forest Random به بالاترین دقت ۹۶,۲۵۷٪ دست یافته است. این یک سیستم شناسایی آنلاین کامل برای فیشینگ است که چندین روش را برای محافظت از کاربران در برابر حمله موثر ترکیب می کند. با این حال، هنوز جای بهبود در عملکرد مدل یادگیری ماشین وجود دارد و تعداد لوگوهایی که طبقه‌بندی کننده لوگو می‌تواند تشخیص دهد بسیار کم است.

Maurya و همکاران (S. Maurya, ۲۰۱۹) یک سیستم ضد فیشینگ را معرفی کردند که حاوی یک پسوند مرورگر وب است. افزونه مرورگر URL فعلی را در زمان واقعی دریافت می کند و ویژگی ها را بر اساس ساختار DOM استخراج می کند، سپس تـــشخیص می دهد که آیا خطر حملات فیشینگ وجود دارد یا خیر و از کاربر درخواست می کند. سرویس تشخیص به سه مرحله تقسیم می شود، یعنی تطبیق لیست سفید، فیلتر لیست سیاه و پیش بینی بر اساس مدل یادگیری ماشین. مرحله پیش‌بینی نشانی اینترنتی را تعیین می‌کند که با معیارها به عنوان پیوند فیشینگ بر اساس ویژگی‌های سطح

کاراکتر مطابقت دارد. به عنوان مثال، هیچ لینکی به صفحه وب وجود ندارد و تعداد لینک‌ها به نام دامنه‌های خارجی از درصد خاصی فراتر می‌رود. چنین قوانینی در برابر مهاجمان آسیب پذیر هستند و برخی از URL های معمولی احتمالاً اشتباه شناسایی می شوند. علاوه بر این، پژوهشگر با ترکیب سه مدل طبقه‌بندی پایه، دقت را بـــــهبود می‌بخشد. شاه و همکاران (B. Shah, ۲۰۲۰) یک افزونه مرورگر مبتنی بر یادگیری ماشینی برای شناسایی URL های فیشینگ ارائه کرد. آنها مدل جنگل تصادفی را با استفاده از مجموعه داده UCI که شامل ۱۱۰۵۵ نمونه با ۳۰ ویژگی نرمال شده است، آموزش دادند. بنابراین، استخراج ویژگی ها بر اساس رشته URL فعلی در یک محیط بلادرنگ ضروری است. در مقاله، نویسندگان ۱۶ ویژگی را استخراج کردند که به خدمات شخص ثالث متکی نیستند. نتایج تجربی نشان می دهد که میزان دقت ۸۹,۶ درصد است که جای پیشرفت زیادی دارد. ساندارام و همکاران (K. M. Sundaram, ۲۰۲۱) یک افزونه مرورگر کروم برای تشخیص وب‌سایت‌های فیشینگ ایجاد کرد. آنها از مجموعه داده های UCI برای آموزش مدل استفاده کردند و مدل آموزش دیده را در یک برنامه افزودنی مرورگر بسته بندی کردند. این مقاله جزئیات پیاده سازی و نتایج را با جزئیات شرح نداده است، و میانگین زمان محاسبه برای تشخیص بلادرنگ URL را ارائه نمی دهد. با این حال، فرآیند استخراج ویژگی به خدمات نام دامنه شـخص ثالث متکی است.

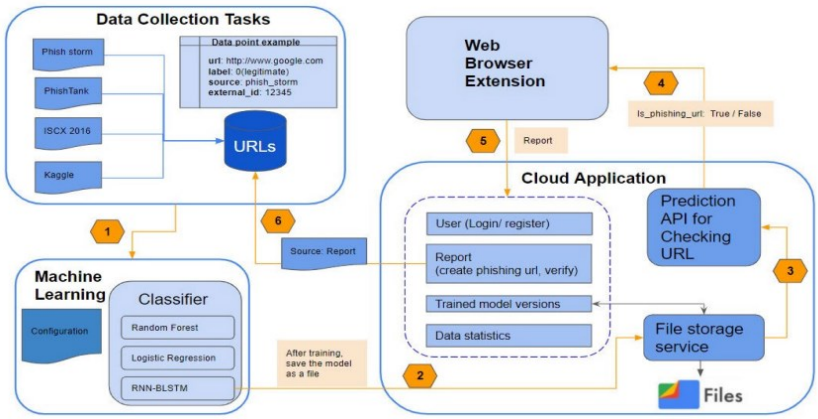
Abiodun و همکاران (O. Abiodun, ۲۰۲۰). وب سایتی را برای تأیید اینکه لینک یک URL فیشینگ است یا خیر، ایجاد کرده است. این آشکارساز توسط زبان برنامه نویسی JAVA و کتابخانه ای به نام JSoup HTML Parser (JHP) پیاده سازی شده است. این راه حل عمدتا به سه مرحله تقسیم می شود. اولین مورد استفاده از JSoup برای تجزیه ساختار DOM وب سایت است که باید شناسایی شود. دوم تجزیه و تحلیل تعداد تگ پیوند <a> از ساختار DOM و تجزیه و تحلیل مقدار ویژگی "href". مقدار مشخصه به عنوان پیوندهای خالی، پیوندهای خارجی و پیوندهای داخلی طبقه بندی می شود. سوم، ماشین‌حساب پیوند نشان‌دهنده‌ای را پیدا کرد که دارای مقداری بین ۰ و ۱ است. وقتی مقدار از ۰،۸ بیشتر شد، URL که باید تأیید شود یک پیوند فیشینگ در نظر گرفته می‌شود. از آنجایی که هیچ مدل یادگیری ماشینی معرفی نشده است، هیچ فرآیند آموزشی وجود ندارد. در این آزمایش، نویسندگان از ۳۰۰ URL برای آزمایش عملکرد ماشین حساب لینک استفاده کردند. نتایج آزمایش نشان داد که آنها به دقت ۹۹،۹۷ درصد و نرخ منفی کاذب ۰،۰۳ دست یافتند. آنها باید از مجموعه داده های آزمایشی بزرگتری برای تأیید این راه حل در آینده استفاده کنند. از تجزیه و تحلیل، قضاوت در مورد خطر فیشینگ تنها با تجزیه و تحلیل ویژگی های برچسب پیوند از روی کد منبع وب سایت، قضاوت نادرستی است و استفاده از این قانون برای مهاجمان برای دور زدن این قوانین آسان است. یک معماری مرورگر وب با یک موتور هوشمند برای تشخیص وب سایت های فیشینگ به نام EPDB در ارائه شده است (M. G. Hr, ۲۰۲۰). در مقایسه با روش‌های معماری سنتی مرورگر وب، EPDB دارای یک مدل یادگیری ماشینی یکپارچه موتور درخشان برای شناسایی در یک محیط بلادرنگ است. آنها از مجموعه داده UCI برای آموزش مدل های یادگیری ماشین استفاده کردند. در فرآیند پیش بینی، قانون چارچوب استخراج اعمال

می شود که می تواند ۳۰ ویژگی یک وب سایت را استخراج کند. نتایج تجربی نشان داد طبقه‌بندی‌کننده جنگل تصادفی با ۹۹/۳۶ درصد بالاترین دقت را به دست آورد. اگرچه دقت داده های تجربی بسیار بالا است، اما این راه حل دارای محدودیت ها و چالش هایی نیز می باشد. اول، توسعه یک مرورگر یک کار بسیار پیچیده است. برخی از عملکردهای مرورگر قبل از اینکه بتوان آنها را برای کاربران تبلیغ کرد، باید با عملکردهای مرورگر بالغ سازگار باشد. علاوه بر این، مجموعه داده برای آموزش مدل تک است و استحکام مدل نیاز به تأیید مجدد دارد. در نهایت، چارچوب استخراج ویژگی مبتنی بر قانون به خدمات شخص ثالث متکی است.

۳_ طراحی چارچوب :

شکل ۱ معماری اجزای چارچوب پیشنهادی ما را نشان می دهد. چهار ماژول از نظر وظایف جمع‌آوری داده، یادگیری ماشین (ML)، برنامه ابری و پسوند مرورگر وب وجود دارد. ماژول جمع آوری داده ها یک برنامه کاربردی وظیفه برنامه ریزی شده مستقل است. ماژول ML برای ماژول های آموزشی استفاده می شود. افزونه مرورگر وب یک محصول سمت مشتری است. برنامه ابری برای مقابله با هشدارهای نادرست و URL های فیشینگ گزارش شده توسط کاربران از افزونه مرورگر وب ساخته شده است. خطوط نارنجی با فلش

در شکل، فرآیند تعامل داده ها را نشان می دهد. فرآیند اصلی این چارچوب عمدتاً به شش مرحله زیر تقسیم می شود: اولین مرحله جمع آوری و یکپارچه سازی داده ها از منابع داده های مختلف است. دوم ترکیب مجموعه های مختلف داده برای آموزش مدل یادگیری ماشین و ذخیره مدل آموزش دیده در یک سیستم فایل. سوم این است که رابط برای پیش‌بینی ریسک فیشینگ، مدل آموزش دیده را برای پیش‌بینی فراخوانی می‌کند. چهارم این است که برنامه افزودنی مرورگر رابط پیش‌بینی را برای انجام شناسایی بلادرنگ و نمایش نتایج تشخیص فراخوانی می‌کند. پنجم این است که کـــــاربران می توانند در صورت مخالفت با نتایج تشخیص، مانند قضاوت نادرست، هشدار از دست رفته، بازخورد بلادرنگ ارسـال کنند. در نهایت، گزارش ارسال شده توسط کاربر از طریق بررسی دستی و استراتژی بررسی خودکار تأیید می شود و نتیجه تأیید با مجموعه داده ها همگام می شود.



شکل۱ــ معماری چارچوب مبتنی بر یادگیری عمیق برای شناسایی یو ار ال های فیشینگ

الف) وظایف جمع آوری داده ها

داده ها هسته اصلی حوزه یادگیری ماشین هستند. کیفیت و کمیت داده ها به طور قابل توجهی بر عملکرد ماژول های مبتنی بر یادگیری ماشین تأثیر می گذارد (N. Gupta, ۲۰۲۱). ماژول جمع آوری داده ها اساس این سیستم است. یک کار جمع آوری داده به دو بخش تقسیم می شود، داده ها را از منابع داده های مختلف به دست می آورند، سپس داده ها را تجزیه و تحلیل و ذخـــیره می کنند. همچنین ما داده ها را از منابع باز مختلف نشان داده شده در جدول زیر جمع آوری کردیم. مجموعه داده (S. Marchal, ۲۰۱٤) طـــوفان Phish شامل ۹۶۰۱۸ URL است: ۴۸۰۰۹ URL قانونی و ۴۸۰۰۹ URL فیشینگ. مجموعه داده ISCX-URL۲۰۱٦ (URL | ۲۰۱٦ | Dataset | Research) شامل ۳۵۳۷۸ URL قانونی و ۹۹۶۵ URL فیشینگ است. ما حدود ۳۵۰۰۰۰ URL خوش خیم را از یک پروژه باز Kaggle بارگذاری

کردیم. (Kumar, ۲۰۲۱) علاوه بر این، در ابتدا ۴۰۰۰۰۰ داده جـــمع‌آوری کردیم و به طور منظم هر روز داده‌های جـــدیدی را از پـــلتفرم Tank Phish (Oct. ۲۰, ۲۰۲۱) دریافت کردیم.

جدول ۱_ منابع داده

Data source	Legitimate	Phishing URLs
Phish storm	۴۸,۰۰۹	۴۷,۹۰۲
Phish Tank	۰	۱۷۸,۴۹۵
ISCX-URL2016	۳۵,۳۷۸	۹,۹۶۵
Kaggle	۳۴۵,۷۳۸	۰

در اینجا ساختار اصلی URL را تجزیه و تحلیل کرده و اطلاعات اولیه مانند پروتکل، دامنه، زیر دامنه، دامنه سطح بالا و مسیر را تجزیه و تحلیل کرده است ([URL Structure, [۲۰۲۰ SEO Best Practices. (۲۰۲۰)). جدول زیر فیلدهای اصلی جدولی به نام URL را نشان می دهد. همچنین داده ها را در یک پایگاه داده رابطه ای ذخیره کرده است، زیرا بـــرای ارائه خدمات داده با خواندن بر اساس SQL انعطاف پذیر و کارآمد است. این سرویس های داده می توانند چندین مجموعه

داده را ترکیب کنند. برای مثال، ۲۰۰۰۰ پیوند فیشینگ از مخزن فیش و ۲۰۰۰۰ پیوند خوب از Kaggle را انتخاب کنید و آنها را در یک مجموعه داده متعادل با ۴۰۰۰۰ نمونه تـــرکیب کنید.

جدول ۲_ ساختار URL جدول پایگاه داده

نام	توضیحات	مدل
url	URL	https://amazom.mhmg.rest/mobile/
Label	۱: فیشینگ ۰: قانونی	1
source	منبع داده	Phish Tank
External_id	شناسه منحصر به فرد منبع داده یکسان	7270002
netloc	netloc	amazom.mhmgmm.rest
GMT_created	تاریخ ایجاد	01:39:40 2021-08-21

ب) یادگیری ماشین

ماژول یادگیری ماشین عمدتاً مسئول آموزش مدل و آزمایش مدل است. در این چارچوب، داده های مدل آموزشی به طور مرتب به روز می شوند و فرآیندهای آموزش و آزمایش همه مدل ها به طور خودکار و منظم راه اندازی می شوند. سیستم پارامترهای هر اجرا و انواع جمع‌آوری داده را ثبت می‌کند و مدل را به سیستم ذخیره‌سازی فایل می‌ریزد. برای افزودن مدل های جدید به ماژول ML انعطاف پذیر است. این تحقیق شش مدل یادگیری ماشین، یعنی رگرسیون لجستیک، ماشین‌های بردار پشتیبان (SVM)، جنگل تصادفی، RNN، RNN-GRU و حافظه کوتاه‌مدت Long-RNN (LSTM) را توسعه داد.

۱) پیکربندی پارامتر

فرآیند پیکربندی پارامتر، پارامترهای مدل را با توجه به فایل پیکربندی مقداردهی اولیه می کند. فایل پیکربندی شامل یک شبکه پارامتر مربوط به هر مدل است و هر پارامتر دارای تعداد گسسته ای از مقادیر است. در فرآیند آموزش مدل، یکی از جایگشت ها و ترکیبات این مقادیر پارامتر برای هر آموزش انتخاب می شود. زمانی که تمام ترکیبات روی مدل اعمال شد و آموزش کامل شد، با مقایسه دقت مدل می توان ترکیب پارامترهای بهینه را به دست آورد.

۲) بارگذاری داده ها

مجموعه داده مورد استفاده برای آموزش مدل از پایگاه داده از طریق سرویس داده به دست می آید. سرویس داده از انتخاب انعطاف‌پذیر ترکیب‌های منبع داده مختلف و مجموعه‌های داده با حجم داده‌های مختلف پشتیبانی می‌کند. هر نمونه داده حاوی یک رشته URL و یک برچسب است که نشان می دهد URL یک پیوند فیشینگ یا پیوند قانونی است. مقادیر برچسب به صورت ۱ و ۰ نرمال می شوند.

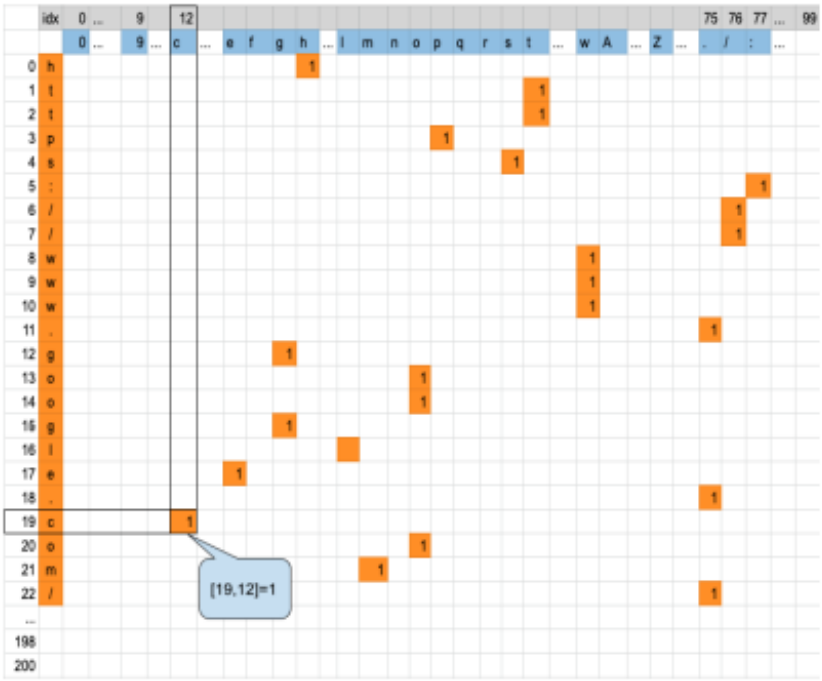
۳) استخراج ویژگی

مجموعه داده مورد استفاده برای آموزش مدل از پایگاه داده از طریق سرویس داده به دست می آید. سرویس داده از انتخاب انعطاف‌پذیر ترکیب‌های منبع داده مختلف و مجموعه‌های داده با حجم داده‌های مختلف پشتیبانی می‌کند. هر نمونه داده حاوی یک رشته URL و یک برچسب است که نشان می دهد URL یک پیوند فیشینگ یا پیوند قانونی است. مقادیر برچسب به صورت ۱ و ۰ نرمال می شوند . ما رشته URL را به عنوان یک سند حاوی معناشناسی در نظر می گیریم و از فناوری پردازش زبان طبیعی (NLP) برای استخراج ویژگی ها استفاده می کنیم. فرآیند استخراج ویژگی، مجموعه‌ای از اسناد متنی را به ماتریسی از تعداد نشانه‌ها تبدیل می‌کند و هر نشانه مخفف یک کلمه است .

در مدل‌های یادگیری ماشینی کلاسیک، فرآیند توکن‌سازی یک رشته URL را به فهرستی از کلمات تبدیل می‌کند. بنابراین، تعداد ویژگی ها برابر با اندازه واژگانی است که با تجزیه و تحلیل داده ها یافت می شود. در مدل‌های یادگیری عمیق، فرآیند توکن‌سازی یک رشته URL را به لیستی از کاراکترها (توکن‌های سطح کاراکتر) تجزیه می‌کند. کاراکترهای موجود در URL از مجموعه کاراکترهای ASCII می آیند. ما متداول ترین ۱۰۰ کاراکتر را به عنوان فرهنگ لغت مجموعه کاراکترها برای این مطالعه انتخاب کردیم. شکل ۲ تمام کاراکترهای مرتب شده و شاخص مربوطه را نشان می دهد. حداکثر طول یک URL ۲۰۸۳ کاراکتر است ([۲۰۲۰ SEO Best Practices] ,URL Structure. (۲۰۲۰)). به دلیل زمان محاسبه مدل یادگیری عمیق و تجزیه و تحلیل داده های آماری مجموعه داده های موجود، حداکثر تعداد کاراکترهای URL را ۲۰۰ تنظیم کردیم. بنابراین، هر URL را می توان به یک ماتریس ۲۰۰*۱۰۰ تبدیل کرد. موقعیت فرهنگ لغت مربوط به هر کاراکتر به عنوان ۱ عـــلامت گـــذاری شده است و مـــقادیر باقی مانده ۰ هستند.

شکل ۲_ فرهنگ لغت کاراکترها با ۱۰۰ کاراکتر اسکی که به طور گسترده در URL ها استفاده می شود

شکل ۳ روند تشکیل یک ماتریس را با استفاده از وب سایت رسمی Google به عنوان مثال نشان می دهد.

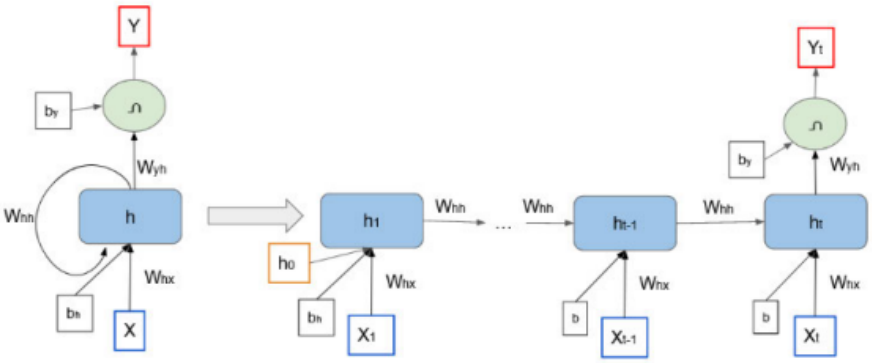


شکل ۳_ فرآیند ایجاد یک ماتریس ویژگی از یک رشته URL و فرهنگ لغت کاراکتر

۴) مدل سازی

این یک راه حل برای تلقی URL به عنوان یک سند و استفاده از جداکننده های کاراکتر برای تجزیه کلمات به عنوان ویژگی است. با این حال، بسیاری از کلمات در URL ها نیز فاقد معنایی هستند. علاوه بر این، تجزیه و تحلیل نتایج سطح کلمه در یک فرهنگ لغت گسترده، زمان محاسبه را کاهش می دهد. بنابراین، ما انتخاب می کنیم که با سطح کاراکتر و کاراکترهای کل URL به عنوان یک دنباله تجزیه و تحلیل کنیم. شبکه عصبی بازگشتی (RNN) یک شبکه عصبی بازخوردی است که حالت های موقت را ذخیره می کند. برای آموزش داده های توالی مناسب است . شکل ۴ یک معماری RNN منظم را نشان می دهد که از یک لایه ورودی، چندین لایه پنهان و یک لایه خروجی تشکیل شده است. در مقایسه با شبکه‌های عصبی مصنوعی پیش‌خور (ANN)، RNN ها معماری منحصر به‌فردی با عملکرد اتصال بین نورون‌ها در لایه‌های پنهان دارند. شکل نشان می دهد که حالت پنهان فعلی با حالت پنهان قبلی و ورودی فعلی مرتبط است. شکل عملکردی حالت پنهان فعلی را می توان به صورت معادله نشان داد. (۱) و (۲). tanh یک تابع غیر خطی است، W وزن بین نورون ها را نشان

می دهد و b بردار بایاس تنظیم است. softmax مقدار خروجی را به عنوان یک تابع فعال سازی محاسبه می کند، همانطور که در معادله نشان داده شده است. (۳)، و مقدار پیش‌بینی مدل مربوط به وضعیت پنهان فعلی است.



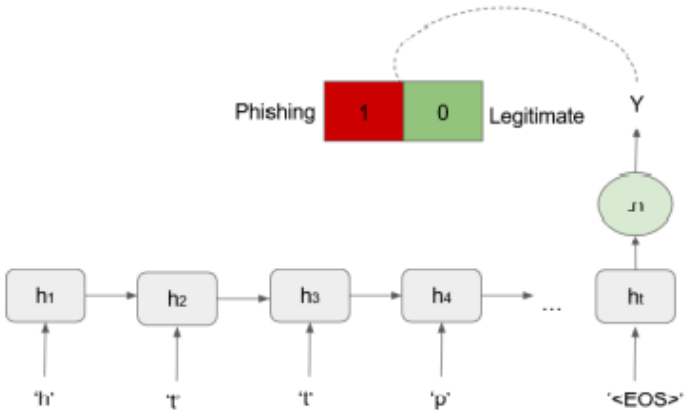
شکل ۴ . معماری یک RNN پایه Whx، Whh، Wyh به ترتیب به معنی ماتریس وزن بین لایه ورودی و پنهان، ماتریس وزن بین دو لایه پنهان، ماتریس وزن بین لایه پنهان و خروجی است

$$\boldsymbol{h_t} = \mathrm{f_w}\ \left(\boldsymbol{h_1} - \mathbf{1}, \boldsymbol{x_1}\right) \quad (۱)$$

$$\boldsymbol{h_t} = \tanh\left(\boldsymbol{w_{hx}x_t} + \boldsymbol{w_{hh}h_{t-1}} + \boldsymbol{b_n}\right) \quad (۲)$$

$$\boldsymbol{y_t} = softmax\left(\boldsymbol{w_{yh^{h_t}}} + \boldsymbol{b_y}\right) \quad (۳)$$

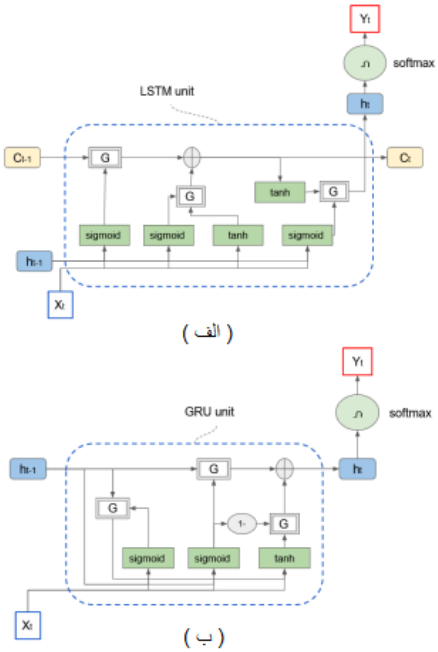
سناریویی که پیوند فیشینگ را شناسایی می‌کند، نوع کار چند به یک است، ورودی داده‌های توالی سطح کاراکتر و خروجی یک دسته است. شکل ۵ ساختار یک لایه پنهان را نشان می دهد.



شکل ۵_ ویژگی‌های سطح کاراکتر در یک مدل RNN برای طبقه‌بندی URL فیشینگ

قبل از آموزش مدل، ساختار مدل ثابت می شود و تابع فعال سازی ایجاد می شود، بنابراین فرآیند آموزش مدل، فرآیند بهینه سازی پارامترهای وزن ها با محاسبه هر خطا است. ابتدا به طور تصادفی ماتریس وزن ها را مقداردهی اولیه کنید، سپس تفاوت بین مقدار واقعی و مقدار پیش بینی شده را محاسبه کنید، سپس از الگوریتم بهینه شده برای یافتن راه حل بهینه برای به حداقل رساندن اختلاف استفاده کنید و در نهایت با محاسبه هر بار گام، هر وزن را تنظیم کنید. بسته به معماری RNN و توابع فعال‌سازی مورد استفاده، معماری اصلی RNN برای مدیریت ورودی‌ها برای توالی‌های طولانی به دلیل آسیب‌پذیری در برابر مشکلات ناپدید شدن گرادیان یا انفجار عملکرد خوبی ندارد (S. Seo, ۲۰۲۰). برای رسیدگی به این موارد، هوکرایتر و اشمیدهابر یک مدل مبتنی بر گرادیان به نام حافظه کوتاه مدت بلند مدت (LSTM) را در سال ۱۹۹۷ معرفی کردند (Schmidhuber, ۱۹۹۷). آنها به جای تابع tanh یک واحد حافظه کوتاه مدت بلند مدت برای محاسبه حالت های پنهان اختراع کردند. واحد LSTM از سه دروازه و دو سلول حافظه تشکیل شده است.

چو و همکاران یک مدل جدید با یک واحد پنهان پیشنهاد کرد که با انگیزه LSTM در سال ۲۰۱۴ ارائه شد (K. Cho, ۲۰۱٤). از آنجایی که واحد مخفی شامل دو گیت برای کنترل و محاسبه حالت پنهان است، این مدل واحد بازگشتی دروازه‌ای GRU نیز نامیده می‌شود. شکل ۶ معماری واحدهای دروازه را نشان می دهد. می توان گفت که شبکه حافظه کوتاه مدت بلند مدت LSTM و واحد بازگشتی دروازه ای GRU دو نسخه پیشرفته RNN هستند. بسیاری از مطالعات و داده‌های تجربی نشان می‌دهند که برای آموزش داده‌های توالی، معماری LSTM و GRU می‌توانند عملکرد بهتری نسبت به معماری پایه RNN داشته باشند (A. Shewalkar, ۲۰۱۹).



(الف)

(ب)

شکل ۶ .(الف) یک واحد LSTM G مخفف یک دروازه است ht .به معنای وضعیت پنهان فعلی و Ct به معنای وضعیت سلول حافظه فعلی است t-۱ .یک بار قبلی است (ب) یک واحد GRU G مخفف یک دروازه است . ht به معنای وضعیت پنهان فعلی است و ht−۱ به معنای وضعیت پنهان قبلی است

۵) بهینه ساز و عملکرد از دست دادن

در فرآیند آموزش مدل، انتخاب یک بهینه ساز و عملکرد تلفات مناسب نیز ضروری است. از میان بسیاری از الگوریتم‌های بهینه‌سازی، Adam را انتخاب کرده‌ایم که یک الگوریتم بهینه‌سازی محبوب و موثر برای یادگیری عمیق است (Ba, ۲۰۱٤). از آنجایی که مشکل یک مسئله طبقه بندی باینری است، از تابع ضرر آنتروپی متقاطع استفاده کردیم (Wookey, ۲۰۲۰) ، که به آن تابع از دست دادن گزارش نیز می گویند. با توجه به سناریوی مشکل فعلی، مقدار پیش‌بینی‌شده خروجی یک عدد ممیز شناور بین ۰ و ۱ است. از دست دادن آنتروپی متقاطع با انحراف احتمال پیش‌بینی‌شده از برچسب واقعی افزایش می‌یابد. اعتقاد بر این است که در مرحله اولیه آموزش با همان میزان یادگیری به سرعت همگرا می شود. ارزش تلفات هر دوره با محاسبه میانگین ارزش تلفات تمام نقاط داده محاسبه می شود .

۶) مدل تخلیه به سیستم فایل

پس از اتمام آموزش مدل، سیستم مدل را در فهرست فایل مشخص شده ذخیره می کند. هر فایل دارای شماره نسخه متفاوتی است و نام فایل مربوط به نسخه است. هنگامی که سیستم در فضای ابری مستقر می شود، از سیستم فایل گوگل برای مدیریت این فایل ها استفاده می کند. سرویس پیش‌بینی می‌تواند این مدل‌ها را مستقیماً بارگیری کند و پیش‌بینی بلادرنگ را برای پیوندهای URL تک یا چـــندگانه انجام دهد و هر تـــماس کمتر از ۲۰۰ میلی‌ثانیه طـــول می‌کشد.

ج) افزونه مرورگر وب

در اینجا یک افزونه مرورگر کروم ایجاد کرده است. از آنجایی که کاربران برنامه افزودنی را در مرورگر کروم نصب کرده‌اند، افزونه بطور خودکار تـــشخیص می‌دهد که آیا URL تازه باز شده در معرض خطر فیشینگ است یا خیر. اگر خـــطری وجود داشته باشد، کاربر به صـــورت تـــعاملی از طـــریق یک کادر بازشو درخواست می‌شود و تـشخیص خـطای بازخورد ورودی ارائـــه می‌شـــود.

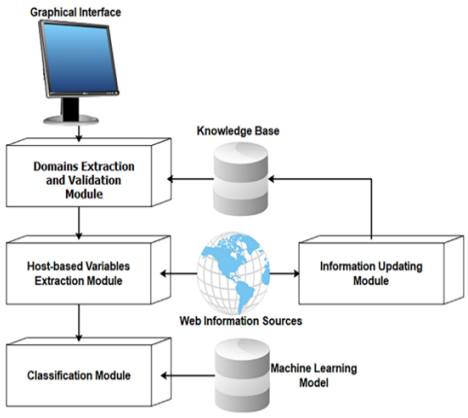
د. کاربرد ابری

هنگامی که یک هشدار نادرست یا هشدار از دست رفته در سرویس پیش بینی رخ می دهد، کاربر می تواند ابتکار عمل را به کار گیرد تا URL شناسایی نادرست فعلی را از پورتال افزونه مرورگر گزارش کند. ما یک وب سایت برای دریافت این گزارش ها ایجاد کرده ایم. پس از ارسال گزارش به سیستم، سیستم یک فرآیند بررسی دستی برای تایید خطرات این URL ها دارد. علاوه بر این، استراتژی‌های حسابرسی خودکار برای بهبود کارایی حسابرسی وجود دارد. پس از تکمیل بررسی، این URL ها به طور منظم با ماژول جمع آوری داده ها هماهنگ می شوند و منبع گزارش می شود. علاوه بر این، وب سایت یک رابط تشخیص برای افزونه های مرورگر ارائه می دهد، از تشخیص چند لایه پشتیبانی می کند و در حال حاضر شامل لیست های سفید، لیست سیاه و مدل های یـــــادگیری ماشینی است.

چارچوب DOCRIW

چارچوب DOCRIW یک پلت فرم نوآورانه است که بر روی شناسایی وب سایت های بالقوه خطرناک متمرکز شده است (Prieto, Fernández-Isabel, & Diego, ۲۰۲۱). بنابراین، می تواند وب سایت ها را به ریسک یا غیر مخاطره آمیز طبقه بندی کند. برای این منظور، دانش را از منابع اطلاعاتی وب بیرونی استخراج می کند و زمـــــانی که هیچ اطلاعاتی

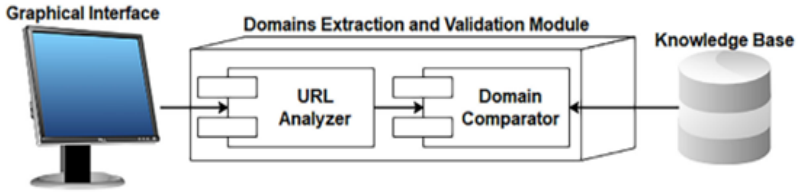
در دسترس نیست، پیش بینی می کند. برای انجام این پیش‌بینی‌ها، DOCRIW اقدامات مشابهی را برای آموزش الگوریتم‌های ML ایجاد می‌کند و از روش‌های بهینه‌سازی برای انتخاب بهترین مدل و پارامترهای مناسب استفاده می‌کند. توجه داشته باشید که رویکرد پیشنهادی به دسترسی مستقیم کاربران به یک وب‌سایت اشاره دارد (یعنی زمانی که کاربران تلاش می‌کنند مورد کلاهبرداری قرار گیرند) . با توجه به معماری کلی سیستم، چهار ماژول اصلی را ارائه می دهد (شکل ۷ را ببینید) : ماژول استخراج و اعتبارسنجی دامنه ها، ماژول استخراج متغیرهای مبتنی بر میزبان، ماژول طبقه بندی و ماژول به روز رسانی اطلاعات. علاوه بر این، این سیستم همچنین دارای یک رابط گرافیکی و دو پایگاه داده است: پایگاه دانش و مدل یادگیری ماشین. رابط گرافیکی وظیفه تعامل با کاربران را بر عهده دارد. پایگاه دانش یک پایگاه داده الستیک سرچ (نصب و اجرا) است که دانش جمع آوری شده از منابع اطلاعات وب را سازماندهی می کند. مدل یادگیری ماشینی شامل طبقه بندی کننده ای هم می باشد .



شکل ۷_ گزیده ای از معماری چارچوب DOCRW

دانش جمع آوری شده از منابع اطلاعاتی وب مدل یادگیری ماشینی شامل کالسی است که قبلا آموزش دیده است. در ادامه بقیه ماژول ها توضیح داده می شوند. الف. ماژول استخراج و اعتبار سنجی دامنه ها این ماژول URL ها را با استخراج دامنه های مربوطه و تجزیه و تحلیل آنها پردازش می کند. برای دستیابی به این وظایف، از ماژول پایگاه دانش

برای به دست آوردن دامنه های مخاطره آمیز قبلی استفاده می کند. ماژول دو جزء را ارائه می دهد: تحلیلگر URL و ارزیابی دامنه شکل ۸ را مشاهده کنید .



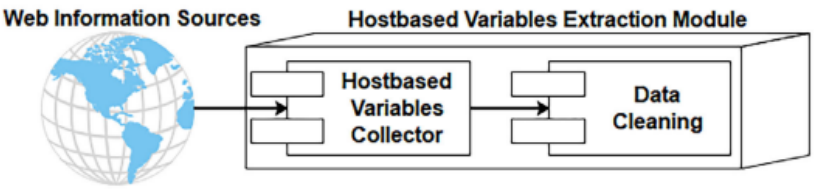
شکل ۸_ گزیده ای از معماری ماژول استخراج و اعتبارسنجی دامنه ها

اولی اطلاعات را از رابط گرافیکی دریافت می کند و در پاسخ به درخواست های کاربران عمل می کند. اطالعات ارائه شده توسط رابط گرافیکی می تواند کل URL ها یا دامنه هایی باشد که قبال از قبل پردازش شده اند. Analyzer URL دامنه پیشنهادی را در هر دو موقعیت ارزیابی می کند. بنابراین، درستی دامنه را بررسی می کند یعنی کد وضعیت برابر با ۲۰۰ است و تغییر مسیرهای احتمالی به صفحات فرود را تشخیص می دهد. در این مورد، تمام صفحات فرود برای تجزیه و تحلیل، استخراج دامنه های مرتبط گنجانده شده است. مؤلفهEvaluator Domains با دامنه های به دست آمده و دامنه های ذخیره شده در پایگاه داده مطابقت دارد. هنگامی که موارد منطبق پیدا می شود، دامنه گزارش شده به عنوان

خطرناک برچسب گذاری می شود. هنگامی که هیچ یک از دامنه ها مطابقت ندارند، ماژول دامنه اصلی را به ماژول استخراج متغیرهای مبتنی بر میزبان می فرستد.

ماژول استخراج متغیرهای میزبان

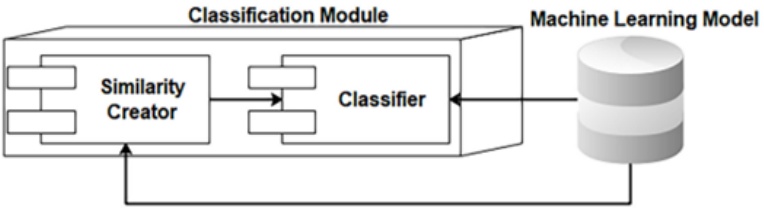
ماژول استخراج متغیرهای مبتنی بر میزبان، متغیرهای جدید مبتنی بر میزبان را از طریق Whois API REST، که بخشی از منابع اطلاعات وب است، جمع آوری می کند. اطلاعاتی در مورد شهر، کشور، تاریخ ایجاد، تاریخ انقضا و ایمیل ارائه می دهد. بنابراین، این ماژول دامنه تحلیل شده را مشخص می کند. با توجه به معماری ماژول (شکل ۹ را ببینید). از دو جزء تشکیل شده است: جمع آوری متغیرهای مبتنی بر میزبان و پاکسازی داده ها. اولی اطلاعات ارائه شده توسط Whois API را مدیریت می کند و یک مجموعه داده را به عنوان خروجی می سازد. مورد دوم به وظیفه پاکسازی می پردازد که نتایج را یکسان می کند. به عنوان مثال، اختصارات کشور با توجه به کد ISO تطبیق داده می شود و عدم تطابق احتمالی بین مقادیر عادی می شود (به عنوان مثال، نام شهر با علائم تاکیدی و همان نام شهر بدون آنها).



شکل ۹_ گزیده ای از معماری ماژول استخراج متغیرهای مبتنی بر میزبان

ماژول طبقه بندی

این ماژول وقتی دامنه ها را در پایگاه دانش یافت نمی شود، به برچسب های مخاطره آمیز یا غیرخطر طبقه بندی می کند. از متغیرهای تولید شده توسط ماژول استخراج متغیرهای مبتنی بر میزبان برای تغذیه مدل یادگیری ماشین استفاده می کند تا یک مقدار پیش بینی شده برای دامنه ها به دست آورد. مدل یادگیری ماشین بر اساس نتایج تجربی انتخاب شده است. مطالعه کامل برای انتخاب عناصر مرتبط با این مدل در ادامه توضیح داده خواهد شد. این مـــدل شامل تعریف شـــباهت بین دامنه ها، یک الگوریتم LR، یک آستانه برای احتمال ارائه شـده توسط الـــگوریتم، و یک مـــجموعه مرجـــع از دامـــنه ها است. شکل ۱۰ را مشاهده کنید .



شکل ۱۰_ گزیده ای از معماری ماژول طبقه بندی

۴_ پیاده سازی نمونه اولیه

اجرای نمونه اولیه کل چارچوب به سه برنامه مستقل تـــقسیم می شود، افزونه مرورگر به صورت مستقل بسته بندی و بر اساس مشخصات توسعه افزونه مرورگر کروم در مرورگر کروم آپلود می شود و توسط پلتفرم کروم بررسی و منتشر خواهد

شد. در توسعه افزونه مرورگر کروم از سه زبان توسعه وب استفاده می‌شود: HTML، جاوا اسکریپت و CSS. برنامه جمع آوری داده ها بر پایه پایتون به عنوان زبان اصلی توسعه است و از وظایف برنامه ریزی شده برای مدیریت وظایف جمع آوری هر منبع داده استفاده می کند. در میان آنها، تانک فیش با استفاده از پرکاربردترین بسته به نام سوپ زیبا، اطلاعات صفحات وب را می خزد. آموزش مدل، خدمات پیش بینی و وب سایت رسمی محصول در یک برنامه ادغام شده است. این برنامه همچنین از Python به عنوان زبان اصلی استفاده می کند و Flask را به عنوان چارچوب وب وارد می کند. آموزش مدل توسط وظایف زمان بندی شده نیز مدیریت می شود. پس از اتمام آموزش، شاخص‌های هسته به ازای عملکرد به صورت بلادرنگ در پایگاه داده MySQL نوشته می‌شود و مدل به سیستم فایل ریخته می‌شود. سرویس پیش‌بینی یک RESTful API است که درخواست‌های POST بلادرنگ را برای به دست آوردن نتایج شناسایی به مشتریان ارائه می‌دهد. وظیفه اصلی وب سایت رسمی پذیرش پیوند مشکوک به فیشینگ ارسال شده توسط کاربر و تعیین خطر پیوند با تأیید دستی و خودکار است.

الف) چارچوب وب

در این بخش از پایتون به عنوان زبان اصلی خود استفاده کرده است، که یک زبان برنامه نویسی سطح بالا در زمینه داده کاوی و یادگیری ماشین است. چارچوب ها و کتابخانه های مختلفی برای زبان پایتون وجود دارد. در سیستم ما، جمع‌آوری داده‌ها، ذخیره‌سازی داده، آموزش مدل، وب‌سایت‌ها و خدمات HTTP همگی توسط کتابخانه‌ها و چارچوب‌های بالغ پشتیبانی می‌شوند. به علاوه دسترسی و استفاده از این بسته ها بسیار ساده و راحت است. با در نظر گرفتن سناریوهای استفاده و عملکرد خواندن و نوشتن، پایگاه داده رابطه ای MySQL را انتخاب کرده است. ابتدا وب سایت دارای مدیریت

کاربر، مدیریت گزارش، مدیریت نسخه مدل و سایر توابع است که نیاز به پایگاه داده رابطه ای دارد. علاوه بر این، مجموعه داده های مورد استفاده برای آموزش مدل به صورت پویا به دست می آید. ترکیب منابع داده های مختلف و حجم داده ها برای تشکیل یک مجموعه داده جدید برای آموزش مدل بسیار انعطاف پذیر است. به عنوان مثال، ما ۲۰۰۰۰۰ URL فیشینگ از مخزن فیش و ۳۴۰۰۰۰ آدرس اینترنتی قانونی از Kaggle به دست آوردیم. یک مجموعه داده متعادل با ۴۰۰۰۰ URL می تواند به طور انعطاف پذیری ترکیب شود، از جمله ۲۰۰۰۰ URL فیشینگ و ۲۰۰۰۰ URL قانونی. ما Flask را به عنوان یک چارچوب وب برای ارائه سرویس HTTP و حفظ وب سایت رسمی وارد کردیم. Flask

یک چارچوب وب سبک وزن است و به راحتی قابل گسترش است. به عنوان مثال، بسته flask-user خدمات مجوز کاربر را ارائه می دهد.

ب) مدل آموزشی

ترکیب منابع داده های مختلف و حجم داده ها برای تشکیل یک مجموعه داده جدید برای آموزش مدل بسیار انعطاف پذیر است. به عنوان مثال، ما ۲۰۰۰۰۰ URL فیشینگ از مخزن فیش و ۳۴۰۰۰۰ آدرس اینترنتی قانونی از Kaggle به دست آوردیم. یک مجموعه داده متعادل با ۴۰۰۰۰ URL می تواند به طور انعطاف پذیری ترکیب شود، از جمله ۲۰۰۰۰ URLفیشینگ و ۲۰۰۰۰ URL قانونی. ما Flask را به عنوان یک چارچوب وب برای ارائه سرویس HTTP و حفظ وب سایت رسمی وارد کردیم. Flask یک چارچوب وب سبک وزن است و به راحتی قابل گسترش است. به عنوان مثال، بسته flask-user خدمات مجوز کاربر را ارائه می دهد.

۱. SCIKIT-LEARN

scikit-learn منبع باز است و به طور گسترده برای تجزیه و تحلیل داده های پیش بینی در زمینه یادگیری ماشین استفاده می شود (Scikit-Learn: Machine Learning in Python—Scikit-Learn ۰,۲۰,۳ .(۲۰۱۹)). در اینجا برای آموزش سه مدل یادگیری ماشین سنتی: رگرسیون لجستیک، جنگل تصادفی، و ماشین بردار پشتیبان، یک کتابخانه یادگیری اسکریپت وارد کرده است.

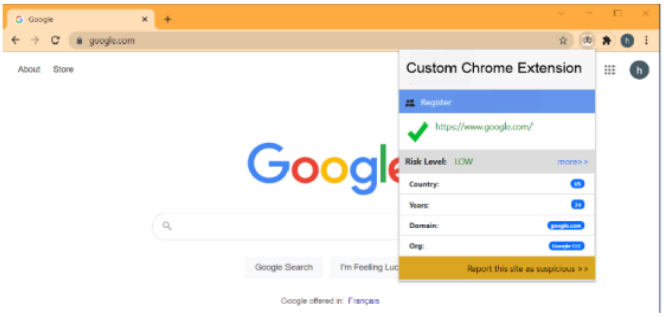
۲. PYTORCH

PyTorch یک چارچوب یادگیری عمیق منبع باز و پلت فرم توسعه است. ما از ماژول مجموعه داده برای ساخت یک مجموعه داده سفارشی به عنوان ورودی برای مدل آموزشی استفاده کردیم. در ساخت مدل‌های یادگیری عمیق، لایه خطی، لایه RNN، لایه GRU و لایه LSTM را وارد کردیم. ما بسته torch.cuda را وارد کردیم که از GPU برای محاسبات موازی استفاده می کند (Torch.Cuda—Pytorch ۱,۹,۱ Documentation, ۲۰۲۱).

ج) افزونه مرورگر وب

توسعه افزونه مرورگر کروم باید کاملاً از دستورالعمل‌های توسعه در (Welcome.Chrome Developers). افزونه کروم دارای پیکربندی برای قرارداد شماره نسخه، معرفی API های داخلی و کنترل مجوز است. تعاملات داده بین ماژول ها پیام داده می شود و به طور موقت در فضای ذخیره سازی Chrome ذخیره

می شود. یک اسکریپت پس‌زمینه به رویدادهای تغییر برگه مرورگر گوش می‌دهد، URL مورد نظر را دریافت می‌کند، با سرویس پیش‌بینی پشتیبان تماس می‌گیرد تا نتیجه را دریافت کند، و در نهایت نتیجه را به یک اسکریپت محتوا ارسال می‌کند. اسکریپت محتوا در درجه اول مسئول ارائه نتایج در صفحه است. علاوه بر این، یک پنجره Html جزئیات را نشان می دهد. شکل ۱۱ نمونه ای از وارد کردن یک URL قانونی را نشان می دهد. در این حالت کاربر صفحه را در یک مرورگر وب بدون اطلاعات اضافی باز می کند. هنگامی که کاربر روی دکمه وصل در سمت راست نوار ابزار کلیک می کند، صفحه بازشو با رشته URL فعلی، سطح خطر و سایر اطلاعات نمایش داده می شود.



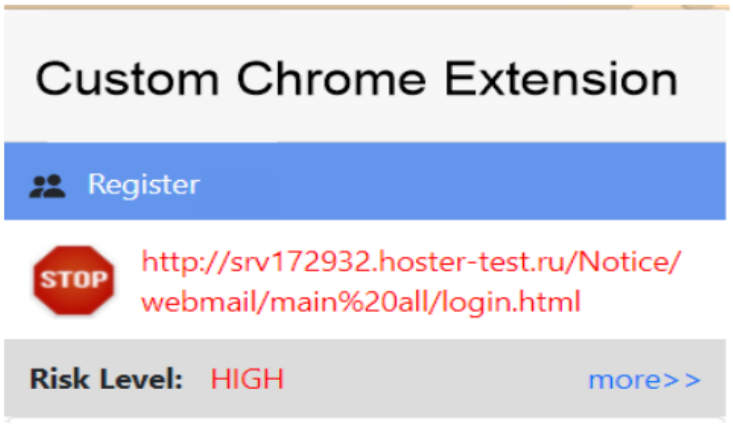
شکل ۱۱_ تصویری از صفحه بازشو افزونه مرورگر کروم

شکل (۱۲) یک مثال را نشان می دهد. هنگامی که URL وارد شده به عنوان یک پیوند فیشینگ شناسایی می شود، یک کادر بازشو با پس زمینه قرمز در صفحه ظاهر می شود که از کاربر می گوید که وب سایت در معرض خطر فیشینگ است. اگر کاربر تأیید کند که URL یک شـــبکه فیشینگ نیست، می تواند روی دکمه هشدار نادرست کلیک کند تا به این هشدار اشتباه پاسخ دهد. شکل (۱۳) سبک و محـــتوای صفحات پاپ آپ را در سایت های پرخطر نـــــشان می دهد.

شکل ۱۲_ تصویری از پنجره پیام هشدار برنامه افزودنی مرورگر کروم هنگام شناسایی URL فیشینگ .

آدرس اینترنتی فعلی-«http://srv۱۷۲۹۳۲.hoster

test.ru/Notice/webmail/main%۲۰all/login.html» است.



شکل ۱۳_ تصویری از صفحه بازشو افزونه مرورگر کروم زمانی که URL فیشینگ را شناسایی کرد.

۵_ نتایج ارزیابی

همه آزمایش‌ها بر روی یک MacBook Pro ۲۰۲۰ با پردازنده چهار هسته‌ای Core Intel ۲ @ i۵ گیگاهرتز با سیستم عامل Sur Big macOS ۱۱,۵,۲ انجام شد. این سرور دارای ظرفیت ذخیره سازی ۵۰۰ گیگابایت است. علاوه بر این، نسبت داده های آزمون ۰,۲ است. در اینجا هفت مجموعه داده، شش مدل داریم و هر مدل پارامترهای متفاوتی دارد. این آزمایش در مراحل زیر برای یافتن سریع مدل بهینه انجام می شود. ابتدا، مدلی را انتخاب کنید که ممکن است از تحلیل نظری عملکرد خوبی داشته باشد. این قسمت مدل GRU را ترجیح می دهد. سپس، مجموعه داده‌ها را با بهترین عملکرد از نتایج تجربی مقایسه می‌کند. دوم، از مجموعه داده های تعیین شده در آزمایش قبلی برای آموزش مدل های مختلف و

مقایسه نتایج استفاده می کند. در نهایت، فراپارامترهای مدل را برای مدل با مجموعه داده بهینه می کند. روش اولیه، شمارش مقادیر گسسته اختیاری پارامترها و انجام ترکیب متقابل برای مقایسه عملکرد همه نتایج تجربی است. ارزیـــابی می کند که آیا یک مدل یادگیری ماشینی معیارهای آماری استاندارد با دقت، یادآوری و دقت بالایی دارد یا خیر . این شاخص ها به روش ساده به دست می آیند محاسبات ریاضی چهار شاخص آماری اتمی بر حسب تعداد موارد مثبت شناسایی شده به درستی (TP)، تعداد نقاط داده منفی شناسایی شده به درستی (TN)، تعداد نقاط داده منفی پیش بینی شده توسط مدل مثبت (FP)، تعداد نمونه های مثبت با برچسب منفی (FN). در مقاله، از امتیاز F۱ برای نشان دادن معنای فراخوانی و دقت استفاده می کنیم. علاوه بر این، در برنامه‌های تشخیص امنیت سایبری، هشدارهای کاذب می‌توانند بر تجربه و اعتماد کاربر تأثیر بگذارند و هشدارهای نشت احتمالاً مستقیماً باعث ضرر کاربر می‌شوند. بنابراین برای اندازه گیری کارایی مدل ها از دقت، F۱، نرخ مثبت کاذب، نرخ منفی کاذب استفاده می کنیم. فرمول های ریاضی این معیارها به صـــورت معادلات (۴)، (۷)، (۸) و (۹) است .

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (٤)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (٥)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (٦)$$

$$F۱ = \frac{2 \times precision \times recall}{precision + recall} = \frac{TP}{TP + \frac{۱}{۲}(FP + FN)} \quad (٧)$$

$$\text{False positive rate} = \frac{FP}{FP+TN} \quad (٨)$$

$$\text{False negative rate} = \frac{FN}{FN+TP} \quad (٩)$$

علاوه بر این، میانگین دقت AP یک متریک پرکاربرد در ارزیابی دقت مدل‌های یادگیری عمیق با محاسبه میانگین مقدار دقت برای ارزش یادآوری بیش از ۰ تا ۱ است. بالاتر بهتر است میانگین دقت متوسط mAP میانگین AP است. مـــعادله (۱۰) منطق محاسبه را نشان می دهد. در این صحنه تعداد کلاس ها دو تا است.

$$mAP = \frac{1}{CLASSES}\sum \frac{TP(c)}{TP(c)+FP(c)} \qquad (10)$$

الف) GRU با مجموعه داده های مختلف

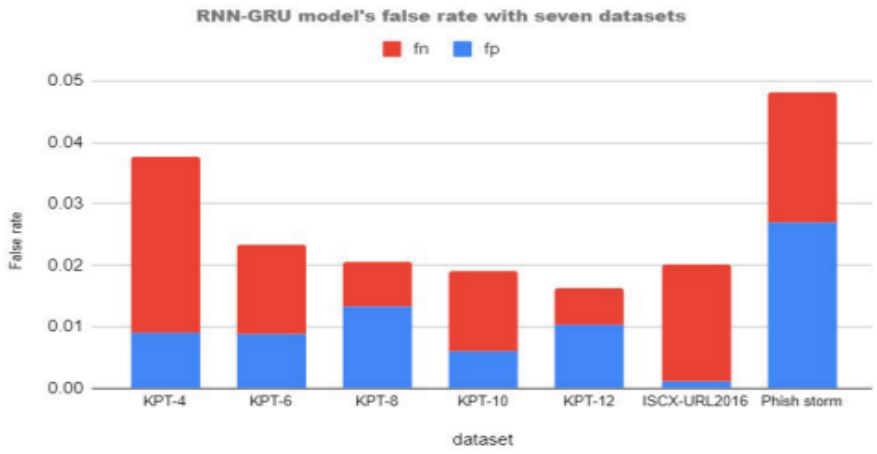
در این آزمایش، مجموعه داده‌های مختلف تغذیه شده با طبقه بندی دیگر GRU را مقایسه کرده است. تعداد دوره ها ۲۰، اندازه دسته ای ۳۲ است، KPT مخفف داده های جمع آوری شده از Kaggle و Tank Phish است، و هر مجموعه داده KPT یک مجموعه داده متعادل است که از همان تعداد URL های فیشینگ و URL های قانونی تشکیل شده است. جدول ۳ شاخص های عملکرد اصلی مدل GRU را با ۸ مجموعه داده نشان می دهد .

جدول ۳_ مدل GRU با مجموعه داده های مختلف

Dataset	accurac	F1	False-positive rate	False negative rate	mAP
Phish storm 95,911 URLs (Legitimate URLs: 48,009; phishing URLs : 47,902)	0.9758	0.9748	0.0269	0.0212	0.96
ISCX:45,343 URLs (Legitimate URLs: 35,375; phishing URLs : 9965)	0.9947	0.9874	0.0011	0.0191	0.986
KPT-4: 40.000 URLs	0.981	0.980	0.0091	0.0285	0.948
KPT-6: 60.000 URLs	0.9882	0.988	0.0089	0.0145	0.967
KPT-8: 80.000 URLs	0.9898	0.9894	0.0134	0.0073	0.973
KPT-10: 100.000 URLs	0.9906	0.9904	0.006	0.0132	0.984
KPT-12: 12.000 URLs	0.9918	0.9915	0.0104	0.0059	0.986

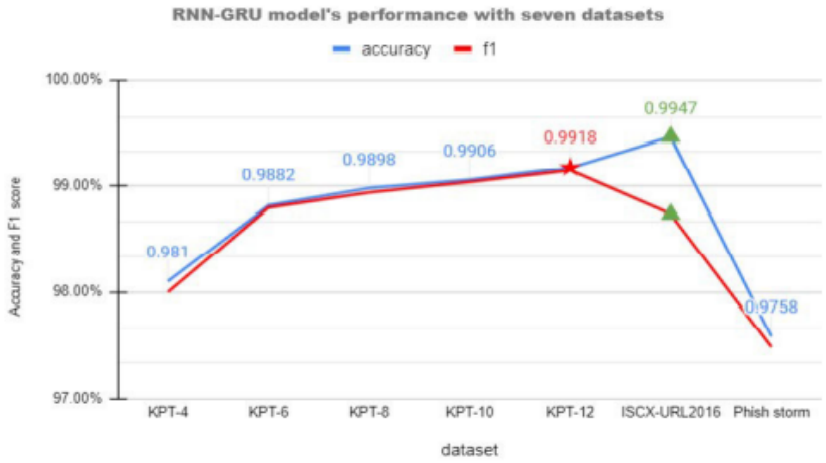
مجموعه داده ISCX بالاترین دقت را به دست آورد. با این حال، امتیاز F۱ کمتر از سه مجموعه داده KPT دیگر است و نرخ منفی کاذب بالا است. به عبارت دیگر، نمونه های قانونی بیشتری به عنوان URL های فیشینگ در طول فرآیند داده های آزمایشی پیش بینی می شوند. این نتیجه احتمالاً به این دلیل است که نقاط داده کافی با عنوان فیشینگ وجـود ندارد. علاوه بر این، نرخ مثبت کاذب و نرخ منفی کاذب را برای اندازه گیری کارایی ترکیب کرده است. از شکل ۱۳، می بینیم که

مدل RNN-GRU با مجموعه داده KPT-۱۲ بهترین عملکرد را دارد. در مجموعه داده های KPT، با افزایش تعداد داده ها، نرخ نادرست به صورت خطی کاهش می یابد.



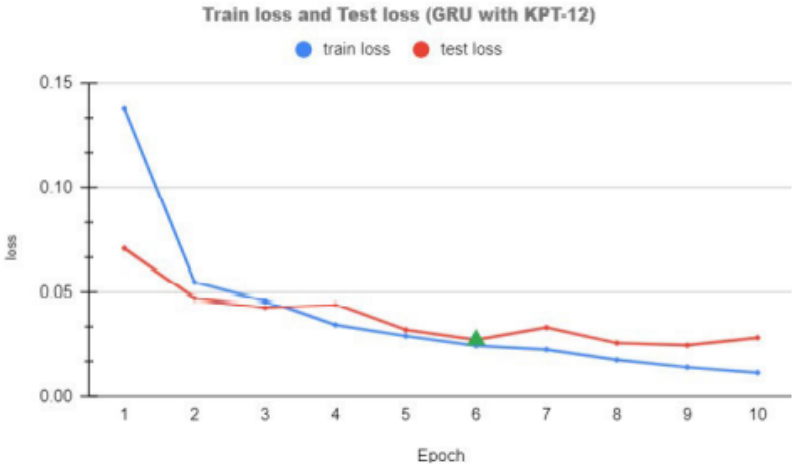
شکل ۱۳_ عملکرد مدل RNN-GRU در آزمایشات با مجموعه داده های مختلف

شکل ۱۴ دقت و F۱ هر مجموعه داده را نشان می دهد. مجموعه داده KPT-۱۲ ۹۹,۱۸٪ دقت و ۹۹,۱۵٪ امتیاز F۱ را به دست آورد. برای تعیین کمیت عملکرد مدل RNN-GRU در هر کلاس، میانگین دقت متوسط (mAP) مجموعه ای از کلاس ها را محاسبه می کنیم. مـیانگین دقت متوسط (mAP) در مــدل های RNN-GRU با مجموعه داده KPT-۱۲ ۰,۹۸۶ است .



شکل ۱۴_ دقت و امتیاز F۱ مجموعه داده های مختلف به یک مدل RNN-GRU اعمال می شود

ارزیابی کارایی یک مدل یادگیری ماشینی، بسته به دقت، ناقص است. در آزمایش‌ها، مرسوم است که در یک مجموعه داده دقت بالایی به دست آورده است و در دیگری عملکرد خوبی نداشته باشیم. همچنین این احتمال وجود دارد که مدل های با دقت بالا داده های جدید را به طور دقیق در یک محیط بلادرنگ پیش بینی نکنند. این شرایط ممکن است به دلیل این واقعیت باشد که بیش از حد برازش قبلا رخ داده است. Overfitting مفهومی در داده کاوی است که تجزیه و تحلیل می کند که آیا یک مدل آموزش دیده می تواند داده های جدید ناشناخته را به طور موثر پیش بینی کند (IBM Cloud Education, ۲۰۲۱). در مدل‌های طبقه‌بندی مبتنی بر یادگیری ماشین، مقایسه خطاها در فرآیند آموزش با خطاهای فرآیند اعتبارسنجی برای دیدن اینکه آیا بیش از حد برازش وجود دارد، همراه با دوره، معمول است. شکل ۹ در این مقاله افت آموزش و افت اعتبار را همراه با دوره ها در مدل RNN-GRU نشان می دهد. یکی از راهبردهای جلوگیری از زیاد تناسب، توقف زودهنگام است (IBM Cloud Education, ۲۰۲۱) ، (Ying, ۲۰۱۹) . همانطور که در شکل ۱۵ نشان داده شده است، دوره برابر با ۶ نقطه مرزی بین underfitting و overfitting است.



شکل ۱۵_ از دست دادن آموزش GRU و از دست دادن آزمایش همراه با دوره ها

ب)مجوعه داده کی پی تی -۱۲ برای طبقه بندی مختلف

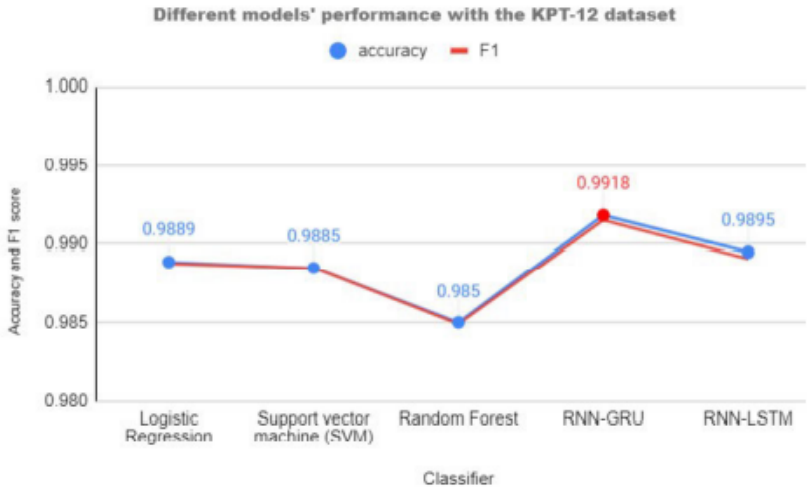
این آزمایش برای اعمال مجموعه داده‌های یکسانی برای مدل‌های مختلف یادگیری ماشینی است که از آن‌ها می‌توان بهترین مدل عملکرد را تحلیل کرد. بسیاری از مـــطالعات نشان داده اند که طبقه بندی کننده جنگل تصادفی بهتر از سایر مدل های طبقه بندی سنتی در تشخیص شبکه های فیشینگ عمل می کند (B. B. Gupta, ۲۰۲۱) ، . بنابراین، ما رگرسیون لجستیک، SVM و جنگل تصادفی را انتخاب کردیم. علاوه بر این، معماری مدل RNN برای آموزش داده های توالی در الگوریتم های یادگیری عمیق ایده آل است. در این آزمایش، معیارهای عملکرد این شش مدل را با هم مـــقایسه کردیم.

جدول ۴ نشان می دهد که RNN-GRU بالاترین دقت ۹۹,۱۸٪ و جنگل تصادفی کمترین نرخ مثبت کاذب ۰,۰۰۴۷٪ را به دست آورد .

جدول ٤ _ مقایسه عملکرد طبقه‌بندی‌کننده‌های مختلف با یک مجموعه داده۱۲-KPT

Classifier	accuracy	F1	False-positive rate	False negative rate
Logistic regression	0.9889	0.9888	0.0081	0.0141
Support vector machine (SVM)	0.9885	0.9885	0.01	0.0129
Random forest	0.985	0.9849	0.0047	0.0253
RNN	0.7412	0.7089	0.1813	0.3372
RNN-GRU	0.9918	0.9915	0.0104	0.0059
RNN-LSTM	0.9895	0.9891	0.0118	0.0089

در این آزمایش، دقت و امتیاز F۱ همه مدل ها بسیار نزدیک بود. این عملکرد را می توان در شکل ۱۶ مشاهده کرد. از آنجایی که دقت مدل RNN زیربنایی کمتر از ۰,۹ است، در شکل نشان داده نشده است. از داده های نتایج سه مدل یادگیری عمیق، اثرات واحد گیت و واحد LSTM بر آموزش داده های توالی یک بار دیگر توضیح داده شده است .



شکل ۱۶_ دقت و امتیاز F۱ در مدل‌های مختلف با مجموعه داده KPT-۱۲

ج) بهینه سازی هایپرپارامتر

از دو نتیجه تجربی بالا، می‌توان نتیجه گرفت که مجموعه داده KPT-۱۲ اعمال شده در مدل RNN-GRU می‌تواند بهترین عملکرد را به دست آورد. آزمایش سوم، بهینه سازی فراپارامترهای مدل برای عملکرد بهتر است. مقادیر اختیاری پارامترها در جدول ۵ فهرست شده است. در مجموع ۱۶۲ ترکیب از مقادیر اختیاری برای همه پارامترها به نوبه خود انجام خواهد شد. از آنجایی که پردازنده گرافیکی رایانه ای که آزمایش را اجرا می کند، از محاسبات موازی پشتیبانی نمی کند و زمان زیادی برای آموزش مدل با مجموعه داده KPT-۱۲ طول می کشد، آزمایش پس از استقرار سیستم در فضای ابری انجام می شود. دسترسی داشته باشید ابزار TensorBoard برای تجسم مقایسه نتایج اجرا و معیارهای عملکرد برای به دست آوردن بهترین ترکیب از پارامترها .

جدول ۵ – مقادیر اختیاری پارامترها

Parameter	Optional value	Default value
Batch size	[32,64,128]	32
epoch	[6,10,20]	10
shuffle	[true,false]	true
Learning - rate	[0.01 , 0.005 , 0.001]	0.001
Layer number	[1,2,3]	2

د) مقایسه

این بخش مدل RNN-GRU را با راه حل های موجود مقایسه می کند که مدل های یادگیری عمیق را برای شناسایی وب سایت های فیشینگ آموزش می دهد. جدول ۶ مقایسه ای از ابعاد مختلف مانند جمع آوری داده ها، مدل ها، شاخص های عملکرد، محدودیت ها را نشان می دهد. در مورد محدودیت‌های پیاده‌سازی راه‌حل پیشنهادی، از آنجایی که هیچ پیوند کوتاهی در مجموعه داده‌های مدل آموزشی وجود ندارد، همه سرویس‌های پیش‌بینی فعلی نمی‌توانند به دقت تشخیص دهند که آیا لینک‌های کوتاه در معرض خطر فیشینگ هستند یا خیر. علاوه بر این، ما ۲۰۰ کاراکتر اول URL را رهگیری کردیم، بنابراین برای آدرس‌های اینترنتی با بیش از ۲۰۰ کاراکتر، بخشی از اطلاعات از بین می‌رود، بنابراین ممکن است بر نتایج تشخیص تأثیر بگذارد. علاوه بر این، روند گزارش بررسی خودکار در حال حاضر بر اساس قوانینی مانند آدرس IP راه دور، اطلاعات مشتری و تعداد دفعاتی که URL ارسال شده است مورد قضاوت قرار می گیرد. این استراتژی می تواند به راحتی توسط مهاجمان فیشینگ به صورت مخرب مورد استفاده قرار گیرد. در آینده، برای پشتیبانی

از نتایج بررسی خودکار، به داده‌های بیشتری نیاز خواهد بود، برای مثال، با به دست آوردن HTML URL فعلی، شناسایی شباهت بین تصویر لوگو و وب‌سایت در لیست سفید، و اینکه آیا یک کادر ورودی در HTML وجود دارد یا خیر .

جدول ۶_ مقایسه مدل پیشنهادی RNN-GRU با سایر راه حل های مبتنی بر یادگیری عمیق

Model or Algorithm	Dataset	Limitations	Accuracy
RNN-GRU	Websites (PhishTank, Kaggle); 120,000 instances:60,000 phishing URLs, 60,000 Legitimate URLs; Character-level features based on URL string.	Short URLs are not supported; URLs of more than 200 characters will lose some of their features	99.18 %
Transfer Learning [44]	Website (Huawei Symantec); 177,417 instances: 36,560 phishing URLs, 14,0857 legitimate URLs; 15 features based on URL string, domain, and sensitive words	Some feature extractions rely on third-party services.	97%
Reinforcement Learning [45]	Website (PhishTank, Yandex Search engine); 73,575 instances:37,175 phishing URLs, 36,400 legitimate URLs. 14 features based on URL string, domain, and HTML	Low accuracy; some feature extractions rely on third-party services.	90.1 %
Convolutional	Websites (PhishTank, PhishStorm,	Some long URLs will lose	97.82 %

۶_ نتیجه گیری و کار آینده

راه‌حل‌های مبتنی بر یادگیری ماشینی بسیاری در سال‌های اخیر برای مقابله با حملات فیشینگ پیشنهاد شده‌اند، اما نتایج در محیط‌های مرور زنده تأیید نشده‌اند و تجزیه و تحلیل و تحقیق درباره محصولات برای تشخیص فیشینگ وجود ندارد. در این مقاله، ما همچنین چارچوبی برای تشخیص فیشینگ در یک محیط مرور زمان واقعی ارائه کردیم. ویژگی های جدید چارچوب عبارتند از: ۱) از داده های حلقه بسته برای ایجاد عملکرد بهتر مدل های یادگیری ماشین استفاده کرده است. یک مجموعه داده برای آموزش مدل اساسی است و داده های با کیفیت بالا می تواند عملکرد یک مدل را بهبود بخشد. داده های بازخورد از کاربران داده های با کیفیت بالا با پیشرفت، دقت و حساسیت هستند. ۲) این سیستم در یک محیط بلادرنگ و بدون تاخیر در حال اجرا است. نتایج پیش بینی با باز شدن صفحه وب نمایش داده می شود.۳) داده های تجربی را می توان

ردیابی کرد. فرآیند آموزش مدل یک کار خودکار است و هر نتیجه اجرا در یک پایگاه داده بلادرنگ ذخیره می شود.۴) همچنین یک افزونه مرورگر را به عنوان محصول مشتری ایجاد کرد که هر کاربر عادی می‌تواند از آن استفاده کند.۵) اجرای سرویس های پیش بینی قابل گسترش است و خدمات تشخیص فردی را می توان ترکیب کرد. برای مثال می توانید یک سرویس فیلترینگ لیست سیاه، سرویس بینایی کامپیوتر را معرفی کنید. ۶) فرآیند استخراج ویژگی در مدل یادگیری عمیق مستقل از خدمات شخص ثالث است. همچنین یک چارچوب جدید به نام DOCRIW ارائه می کند که مبنتی بر پایگاه دانش می باشد، که دامنه ها را به عنوان پرخطر و غیرخطر طبقه بندی می کند. برای این منظور، از منابع اطلاعاتی وب برای جمع آوری دانش خاص از حوزه های بالقوه مخاطره آمیز استفاده می شود. این عملکرد با یک طبقه بندی ML بر اساس الگوریتم LR تکمیل می شود. کالسیور از طریق ۱ آموزش دیده است. ۵۰۰ دامنه دارای برچسب. نتایج امیدوارکننده ای از طریق آزمایش های متعددی که برای اثبات زنده بودن این پیشنهاد انجام شده است، به دست آمده است. حتی اگر چارچوب DOCRIW یک سیستم کامل را نشان دهد، پیشرفت های آینده می تواند برای افزایش عملکرد کلی اعمال شود. پایگاه داده ای که شامل دامنه هایی است که به عنوان ریسک RiskyDomainsDB برچسب گذاری شده اند، فقط برخی از دامنه ها را شامل می شود. بنابراین، بهبود بیشتر ماژول پیش بینی الزامی است. این ماژول ابتدا ماتریس های شباهت را محاسبه می کند تا دامنه ها را به عنوان پرخطر یا غیرخطر طبقه بندی کند. شش ویژگی برای تعیین شش شباهت مختلف برای اندازه گیری شباهت بین دامنه ها استفاده می شود. مورد اول از فاصله نرمال شده shtein-Leven برای نام دامنه استفاده می کند. سایر معیارهای شباهت از مکاتبات بین شهرها، کشورها، تاریخ ایجاد، تاریخ انقضا و ایمیل ها تعیین می شوند. شباهت سراسری از طریق ترکیب وزنی از تمام این شباهت های فردی ایجاد می شود، که در آن وزن ها نمایانگر میزان هر ویژگی هستند. اولین محدودیت توسط معیار شباهت لونشتاین به دلیل نادیده گرفتن مفاهیم معنایی ارائه شده است. همچنین Levenshtein به نتایج بهتری از Cosine Similarity C IDF-T رسیده است. با این حال، چندین معیار شباهت معروف را

می تـوان آزمـــــــایش و مـــــــقایسه کرد به عنوان مثال، فاصله ویرایش، شباهت اسمیت- واترمن، شباهت جارو-وینکلر، یا شباهت مونگ-الکان تکنیک های NLP همچنین می توانند برای افزودن مؤلفه معنایی به

ماتریس های شباهت مورد بررسی قرار گیرند، حتی اگر نمی توانند سیستم را غنی کنند. عالوه بر این، تنها از مـــــتغیرهای مبتنی بر میزبان ve برای ارتقای شباهت استفاده شده است زیرا بقیه آنها هیچ اطلاعاتی را در اختیار مدل ها قرار نمی دهند. این احـــــتمال وجود دارد که تنها شش متغیر کافی نباشد، در نظر گرفـتن ویژگی های جدیدی که می تواند اطلاعات مرتبط را برای دامنه ها فراهم کند، می تواند جالب باشد. با توجه به الگوریتم های ML، تنها ۱ ;۵۰۰ دامنه که قبلا توسط کارشناسان برچسب گذاری شده بود استفاده شده است. این محدودیت حجم نمونه را می توان با بازآموزی طبقه بندی LR با آن دسته از حوزه هایی که احتمال خطر یا غیر مخاطره آمیز بودن آنها باالست کاهش داد و همچنین می توان تکنیک های یادگیری تقویتی را نیز گنجاند. توجه داشته باشید که در دنیای واقعی، تعداد دامنه های غیر ریسکی بسیار بیشتر از دامنه های مخاطره آمیز است. در این مقاله مجموعه ای از حوزه های غیرریسکی به همان اندازه که مجموعه حوزه های مخاطره آمیز در نظر گرفته شده است. در آینده، رویکردهای مختلفی برای مقابله با مشکالت نامتعادل، مانند جریمه کردن طبقه بندی اشتباه به روش های مختلف، در نظر گرفته خواهد شد. در نهایت، تـــــحقیقات بیشتر با سـایر طبـقه بندی ها و ترکیب ها و همـــــــچنین روش های گـروهی جـالب خواهد بود .

References

١. (٢٠١٩). Scikit-Learn: Machine Learning in Python—Scikit-Learn ٠,٢٠,٣. (n.d.). Retrieved from https://scikit-learn.org/stable/index.html

٢. (٢٠٢٠). URL Structure, [٢٠٢٠ SEO Best Practices]. (n.d.). Retrieved from https://moz.com/learn/seo/url

٣. A. Shewalkar, D. N. (٢٠١٩). Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU.

٤. B. B. Gupta, K. Y. (٢٠٢١). A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment.

٥. B. Shah, K. D. (٢٠٢٠). Chrome Extension for Detecting Phishing Websites.

٦. Ba, D. P. (٢٠١٤). Adam: A Method for Stochastic Optimization.

٧. Cho, S.-J. B.-B. (٢٠٢١). Deep Character-Level Anomaly Detection Based on a Convolutional Autoencoder for Zero-Day Phishing URL Detection .

٨. D. N. Atimorathanna, T. S. (٢٠٢٠). NoFish; Total Anti-Phishing Protection System.

٩. IBM Cloud Education. (٢٠٢١). Retrieved from https://www.ibm.com/cloud/learn/overfitting

١٠. K. Cho, B. v. (٢٠١٤). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.

١١. K. M. Sundaram, R. S. (٢٠٢١). Detecting phishing websites using an efficient feature based machine learning framework.

١٢. Kumar, S. (٢٠٢١). Malicious and Benign URLs. Retrieved from https://www.kaggle.com/

١٣. M. G. Hr, M. V. (٢٠٢٠). Development of anti-phishing browser based on random forest and rule of extraction framework.

١٤. Mohammad, R. T. (٢٠١٤). *Predicting phishing websites based on self-structuring neural network*.

١٥. N. Gupta, S. M.-h. (٢٠٢١). Data Quality for Machine Learning Tasks.

١٦. O. Abiodun, A. S. (٢٠٢٠). Linkcalculator – An Efficient Link-Based Phishing Detection Tool.

۱۷. (Oct. ۲۰, ۲۰۲۱). Retrieved from https://www.phishtank.com/ . https://www.phishtank.com/phish_archive.php

۱۸. Prieto, J. C., Fernández-Isabel, A., & Diego, I. M. (۲۰۲۱). Knowledge-Based Approach to Detect Potentially Risky Websites.

۱۹. S. Marchal, J. F. (۲۰۱٤). PhishStorm: Detecting Phishing With Streaming Analytics.

۲۰. S. Maurya, H. S. (۲۰۱۹). Browser Extension based Hybrid Anti-Phishing Framework using Feature Selection.

۲۱. S. Seo, C. K. (۲۰۲۰). Comparative Study of Deep Learning-Based Sentiment Classification.

۲۲. Safara, M. S. (۲۰۲۱). ISHO: improved spotted hyena optimization algorithm for phishing website detection.

۲۳. Schmidhuber, S. H. (۱۹۹۷). Long Short-Term Memory .

۲٤. Torch.Cuda—Pytorch ۱,۹,۱ Documentation. (۲۰۲۱). Retrieved from https://pytorch.org/

۲٥. Triantaphyllou, H. N. (۲۰۰۸). The Impact of Overfitting and Overgeneralization on the Classification Accuracy in Data Mining.

۲٦. URL | ۲۰۱٦ | Dataset | Research. (n.d.). Retrieved from https://www.unb.ca/cic/datasets/url-۲۰۱٦.html

۲۷. Welcome.Chrome Developers. (n.d.). Retrieved from https://developer.chrome.com/docs/extensions/mv۳/

۲۸. Wookey, Y. H. (۲۰۲۰). The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling.

۲۹. Ying, X. (۲۰۱۹). An Overview of Overfitting and its Solutions.