به نام خدا

عنوان پروژه: شناسایی حملات فیشینگ[1]با استفاده از الگوریتم های یادگیری ماشین[2]

تاریخ:۲۱-آبان_۱۴۰۰

نوع مسئله:یادگیری بانظارت(طبقه بندی)

کارهای گذشته:

(مهمت کارکمز و همکاران،۲۰۲۱)،دسته بندی لینک های فیشینگ[3]با استفاده از شبکه های عصبی عمیق رو دیتاست لینک های پر خطر

با توجه به روند رو به رشد اینترنت، تعداد رایانه های متصل به اینترنت روز به روز در حال افزایش است. تقریبا همه شرکت ها در حال انتقال اعمال اصلی خود از دنیای واقعی به دنیای سایبری هستند،اگر چه آن موضوع باعث افزایش فروش آنها میشود میتواند آسیب پذیری های زیادی مانند حملات سایبری برای این شرکت ها را درپی دارد به ویژه با ساختار ناشناس اینترنت.فیشینگ یکی از انواع محبوب حملات سایبری است که از ناآگاهی کاربر برای آسیب رساندن به آن ها سو استفاده می شود.در برخی از پزوهش ها از سیستم های تشخیص مبتنی برقانون بعنوان یک مکانیسم پیشگیری استاتیک و سیستم های مبتنی بر یادگیری ماشین به عنوان یک مکانیسم پیشگیری پویا کمک میگیرد.در مقاله نام برده شده نویسندگان یک سیستم تشخیص فیشینگ مبتنی بر شبکه های عصبی عمیق[4]را با تجزیه و تحلیل نشانی اینترنتی وب سایت های مشکوک پیاده سازی کرده اند.اگر چه در تمام تحقیقاتی قبلی که توسط محققان دیگر انجام شده مجموعه داده های مورد استفاده از منابع مختلفی جمع آوری شده بودند وکه وب سایت های مجاز و فیشینگ در آنها مشخص است.اما نویسندگان مقاله نام برده ابتدا یک مجموعه داده پرخطر ایجاد می کنند که فقط شامل وب سایت

[1] phishing
[2] Machine learning
[3] Phishing
[4] Deep Neural Network (DNN)

های مشکوک است که به سایت فیشتانک[5]گزارش شده اند.تحقیقاتی تجربی نویسندگان مقاله نام برده نشان می دهد که سیستم پیشنهادی توسط نویسندگان مقاله نام برده کارآیی بسیار خوبی از نظر دقت و زمان اجرا ارائه می دهد.

(او.اسماعیل و همکاران ،۲۰۱۰)،کاهش خطر حملات فیشینگ با استفاده از ابزار های هوش مصنوعی

در مقاله نام برده شده یک پژوهش جامع به همراه تجزیه و تحلیل در مورد فیشینگ، ویشینگ[6] اسمیشینگ[7]برای بهره برداری از دانش در پیاده سازی یک ابزار هوشمند برای شناسایی و محافظت در مقابل موارد ذکر شده ارائه شده است. این یک مشکل جدید مهندسی اجتماعی است که زندگی روزمره ما را آسیب پذیر و دشوار می کند. تحقیق نام برده شده به ویژه بر روی فیشینگ از طریق ایمیل متمرکز است زیرا ، به طور مستقیم پیامدهای جدی تری در مقایسه با سایر روش ها در رابطه با معاملات مالی دارد. شایان ذکر است که امن نمودن حجم بسیار زیاد معاملات آنلاین بسیار چالش برانگیز است زیرا روزانه چندین روش برای نقض حریم خصوصی افراد به منظور سرقت اعتبار آنها ابداع می شود. هزینه این نوع حملات بیش از میلیون ها دلار به صورت سالانه است. ابزارهای زیادی برای حل این مشکل ارائه شده است. اما متأسفانه، این معضل هنوز پا برجاست. در مقاله نام برده شذه توسط نویسندگان یک روش برای توسعه یک ابزار هوشمند برای مقابله با این خطر پیشنهاد شده است.

(یزدان احمد السریرا وهمکاران،۲۰۲۰)،الگوریتم های فرایادگیری و درختان تصمیم اضافی برای شناسایی صفحات فیشینگ

فیشینگ نوعی حمله مهندسی-وب اجتماعی[8]در فضای مجازی است که در آن مجرمان اطلاعات ارزشمند کاربران ناآگاه را به سرقت می برند. اقدامات متقابل موجود در قالب نرم افزار ضد فیشینگ

---

[5] PhishTank
[6] Vishing
[7] Smishing
[8] social web-engineering

و روش های محاسباتی برای شناسایی فعالیت های فیشینگ اثبات شده است. با این حال، هکرها روش های جدیدی را برای خنثی سازی این اقدامات متقابل به کار می گیرند. با توجه به ماهیت تکامل یافته حملات فیشینگ، نیاز به اقدامات متقابل جدید و کارآمد بسیار مهم احساس می شود زیرا تأثیر حملات فیشینگ اغلب فاجعه بار است. طرح های هوش مصنوعی سنگ بنای روش های مقابل مدرنی است که برای کاهش حملات فیشینگ استفاده می شود. اقدامات متقابل یا روش های مقابله ای فیشینگ مبتنی بر هوش مصنوعی دارای نواقصی به ویژه بالا بودن میزان هشدار کاذب و عدم توانایی تفسیر نحوه عملکرد بیشتر روش های فیشینگ است.در مقاله نام برده شده نویسندگان چهار مدل فرا آموز[۹]آدابوست اکسترا تری،بگینگ اکسترا تری،[۱۱]روتیشن فارست اکسترا تری،[۱۲]لجیت بوست اکسترا تری،[۱۳]بر مبنای الگوریتم طبقه بندی اکسترا تری توسعه داده اند. مدل ها توسط نویسندگان این مقاله روی مجموعه داده های وب سایت های فیشینگ آموزش داده شده وعملکرد آنها مورد ارزیابی قرار گرفته است.دقت تشخیص مدل ها بالای ۹۷درصد با نرخ مثبت کاذب ناچیز ۰،۰۲۸بوده است.علاوخ بر این مدل های پیشنهاد شده توسط نویسندگان مقاله نام برده شدهدرتشخیص حملات فیشینگار مدل های مبتنی برای یادگیری ماشینی برتری دارند.از این رو نویسندگان مقاله نام برده شده مدل های فرا آموز را برای تشخیص حملات قیشینگ اتخاذ نموده اند.

(هانگ لی . همکاران،۲۰۱۷)، یوارال نت[۱۴]:یادگیری لینک های نماینده با یادگیری عمیق برای شناسایی لینک های مخرب

[۹] meta-learner
[۱۰] AdaBoost-Extra Tree (ABET)
[۱۱] Bagging - Extra tree (BET)
[۱۲] Rotation Forest - Extra Tree (RoFBET)
[۱۳] LogitBoost-Extra Tree (LBET)
[۱۴] URLNET

نشانی های مخرب محتوای ناخواسته را میزبانی می کنند و برای ارتکاب جرایم اینترنتی استفاده می شوند. تشخیص به موقع آنها ضروری است. به طور سنتی، این کار با استفاده از لیست های سیاه انجام می شود، که نمی تواند جامع باشد و نمی تواند نشانی های مخرب تازه ایجاد شده را شناسایی کند. برای حل این مسئله، در سال های اخیر شاهد تلاش های زیادی برای شناسایی لینک های مخرب با استفاده از یادگیری ماشین بوده ایم. محبوب ترین و مقیاس پذیرترین روش ها با استخراج ویژگی های بگ اف وردی[۱۵]از خصوصیات رشته واژه نشانی ها استفاده می کنند و به دنبال آن مدل های یادگیری ماشین مانند ماشین بردار پشتیبان[۱۶]استفاده می شود. همچنین ویژگی های دیگری نیز وجود دارد که توسط متخصصان برای بهبود عملکرد پیش بینی مدل طراحی شده است. این رویکردها از چندین محدودیت رنج می برند:شماره یک- عدم توانایی در دستیابی موثر معنایی و الگوهای پی در پی در رشته نشانیوبسایت ها -شماره دو-نیاز به انجام مهندسی ویژگی های به صورت دستی و شماره سه- عدم توانایی در مدیریت ویژگی های غیبی و تعمیم دادن آن ها برای آزمایش داده ها. برای رفع این چالش ها نویسندگان مقاله نام برده یو ار ال نت را پیشنهاد می دهند، یک چارچوب یادگیری عمیق برای یادگیری یک جاسازی نشانی های غیرخطی که نشانی های مخرب را مستقیما از نشانی های معمولی تشخیص می دهد. به طور خاصنویسندگان مقاله نام برده شده شبکه های عصبی کانولوشنی[۱۷]را به هر دو گروه حروف و کلمات رشته نشانی ها اعمال می کنند تا نشانی تعبیه شده را در یک چارچوب مشترک بهینه سازی شده یاد بگیرد. این رویکرد به مدل اجازه می دهد تا انواع مختلفی از اطلاعات معنایی را بدست آورد، که توسط مدل های موجود امکان پذیر نبود. نویسندگان مقاله نام برده شده همچنین تعبیه پیشرفته کلمات را پیشنهاد می دهیم تا مشکل بسیاری از کلمات نادر مشاهده شده در این کار را برطرف کنیم. نویسندگان مقاله نام برشده آزمایش های گسترده ای را روی یک مجموعه داده در مقیاس بزرگ انجام می دهند

---

۱۵ Bag-of-words
۱۶ SVM
۱۷ Convolutional

که این آزمایشات عملکرد قابل توجهی نسبت به روش های موجود نشان می ده.نویسندگان مقاله نام برده شده برای ارزیابی عملکرد اجزای مختلف یوارال نت مطالعات فرسایش را انجام می دهند.

(دایساکی میامتو و همکاران،۲۰۰۹)،ارزیابی روش های مبتنی برا یادگیری ماشینی برای شناسایی سایت های فیشینگ

در مقاله نام برده شده، نویسندگان این مقاله عملکرد روش‌های مبتنی بر یادگیری ماشین را برای تشخیص سایت‌های فیشینگ ارائه می‌دهند. نویسندگان این مقاله از ۹ تکنیک یادگیری ماشینی استفاده می‌کنند که عبارتند از: آدابوست[۱۸]،بگینگ[۱۹]،ماشین‌های بردار پشتیبان[۲۰]،درخت طبقه‌بندی و رگرسیون[۲۱]، لجستیک رگرسیون[۲۲]، جنگل‌های تصادفی[۲۳]، شبکه‌های عصبی[۲۴]، نایو بایز[۲۵] و درخت‌های رگرسیون افزایشی بیزی[۲۶].نویسندگان این مقاله اجازه می‌دهند که این تکنیک‌های یادگیری ماشینی، شیوه‌های اکتشافی را با هم ترکیب کنند، و همچنین اجازه می‌دهند که روش‌های تشخیص مبتنی بر یادگیری ماشین، سایت‌های فیشینگ را از سایرین متمایز کنند. نویسندگان این مقاله مجموعه داده‌های خود را که متشکل از ۱۵۰۰ سایت فیشینگ و ۱۵۰۰ سایت قانونی است تجزیه و تحلیل می‌کنند، نویسندگان مقاله نام برده شده داده های خود را با استفاده از روش‌های تشخیص مبتنی بر یادگیری ماشین طبقه‌بندی می‌کنند، و عملکرد هر دسته بند را اندازه‌گیری می‌کنند. نویسندگان این مقاله از معیار اندازه گیری اف وان[۲۷]،نرخ خطا، و مساحت تحت منحنی اراو سی[۲۸] عنوان معیارهای عملکرد هر الگوریتم دست بند استفاده کرداند. بالاترین مقدار اف وان برابر با ۰,۸۵۸۱، کم‌ترین میزان خطا برابر با ۱٤,۱۵ درصد و بالاترین میزان ای یو سی برابر با ۰,۹۳٤۲

---

۱۸ AdaBoost
۱۹ Bagging
۲۰ Support Vector Machines
۲۱ Classification and Regression Trees
۲۲ Logistic Regression
۲۳ Random Forest
۲۴ Neural Network
۲۵ Naive Bayes
۲۶ Bayesian Additive Regression Trees
۲۷ f1 measure
۲۸ AUC

می‌باشد که همگی در مورد آدابوست مشاهده شده اند. همچنین نویسندگان این مقاله مشاهده کرده اند که ۷ روش از ۹ روش تشخیص مبتنی بر یادگیری ماشین عملکرد بهتری نسبت به روش تشخیص سنتی دارند.

(یوسف عراقی و همکاران،۲۰۱۳)،شناسایی فیشینگ: بررسی ادبیات[۲۹]

درمقاله نام برده نویسندگان به بررسی مقالات گوناگون در مورد روش های تشخیص حملات فیشینگ می‌پردازند. حملات فیشینگ، آسیب‌پذیری‌های موجود در سیستم‌ها را به واسطه عامل انسانی مورد هدف قرار می‌دهند. بسیاری از حملات سایبری از طریق مکانیزم‌هایی گسترش می‌یابند که از نقاط ضعف موجود در کاربران بهره‌برداری می‌کنند، که این مسئله کاربران را به ضعیف‌ترین عنصر در زنجیره امنیتی تبدیل می‌کند. مشکل فیشینگ گسترده است و هیچ راه حلی برای کاهش موثر تمام آسیب‌پذیری‌های ناشی از آن وجود ندارد، بنابراین تکنیک‌های متعددی برای کاهش حملات خاص اجرا می‌شوند. هدف مقاله نام برده شده بررسی بسیاری از تکنیک‌های کاهش حملات فیشینگ است که اخیرا پیشنهاد شده‌اند. در مقاله نام برده دسته‌های مختلفی از تکنیک‌های سطح بالا که جهت کاهش تعداد حملات فیشینگ وجود دارد ارائه شده‌است، مانند: تشخیص، دفاع تهاجمی، اصلاح، و پیش‌گیری، که نویسندگان این مقاله معتقداند شناسایی حملات فیشینگ حیاتی ترین روش برای کاهش تعداد حملات فیشینگ می باشد.

(جیان کاو و همکاران،۲۰۱۶) ،شناسایی لینک های مخرب بر اساس باز ارسال در شبکه های اجتماعی آنلاین

در سال های اخیر شبکه های اجتماعی آنلاین[۳۰]مانند فیسبوک ،توئیتر و...درمیان کاربران اینترنت بسیار محبوب شده اند.متاسفانه مهاجمان از آنها برای مخفی کردن نشان های مخرب نیز استفاده میکنند.با توجه به اهمیت شناسایی نشانی مخرب در شبکه های اجتماعی آنلاین،چندین راه توسط

---

[۲۹] A Literature Survey

[۳۰] OSN

اپراتور های شبکه های اجتماعی آنلاین،شرکت های امنیتی و محققان دانشگاهی ارائه شده است،بیشتر این راه حل ها از روش های یادگیری ماشین برای آموزش مدل های طبقه بندی براساس انواع مختلف مجموعه ویژگی ها استفاده می کنند.با این حال بیشتر این مدل های آموزش دیده بی اثر هستند زیرا ویژگی های انتخاب شده آنها معمولی است.در مقاله نام برده شده نویسندگان به ویژگی های مبتنی بر باز ارسال تمرکز کرده اندزیرا ارتباط ویژه ای بین رفتار ارسال و انتشار نشانی های مخرب وجود دارد.در ابتدا نویسندگان مقاله یک تجزیه و تحلیل جامع از مجموعه ویژگی های نشانی های معمولی انجام داده اند.سپس برخی از ویژگی های مبتنی بر ارسال و باز ارسال را طراحی کرده اند و چندین ویژگی مبتنی بر نمودار را انتخاب کرده اند تا با ویژگی های قبلی ترکیب شوند تا یک مدل شناسایی لینک های مخرب را با آن ها آموزش دهند نویسندگان مقاله نام برده سیستم مورد نظر خود را با استفاده از حدود ۱۰۰۰۰۰پیام اصلی از سیناویبو[۳۱]جمع آوری شده ارزیابی میکنند.میزان دقت بالا و نرخ مثبت کاذب پایین نشان می دهد که ویژگی های مبتنی بر ارسال و باز ارسال در شناسایی نشانی های مخرب در شبکه های احتماعی آنلاین بسیارموثرتراز سایر ویژگی های مرسوم هستند. از نظر دانش نویسندگان مقاله نام برده شده این مقاله اولین مقاله ای می باشد که ویژگی های مبتنی بر ارسال و باز ارسال را در شبکه های اجتماعی آنلاین تجزیه و تحلیل می کند و سهم ارزشمندی در تحقیقات این زمینه ارائه می دهد.

(انی ویژالی و همکاران،۲۰۱۸)،پی ای دی-ام ال[۳۲]شناسایی ایمیل های فیشینگ با اشتفاده از تکنیک های یادگیری ماشینی کلاسیک

در دوران مدرن، کلیه خدمات به صورت آنلاین صورت می پذیرد و همه از آن برای سرعت بخشیدن به فعالیت های روزمره خود استفاده می کنند. این شامل فعالیت های اجتماعی و همچنین مالی است که شامل استفاده از اطلاعات حساس برای انجام وظیفه مورد نظر است. با افزایش استفاده از چنین امکاناتی اهمیت ایمن سازی داده های مورد استفاده برای این اقدامات مطرح می شود. طی

[۳۱] Sina Weibo
[۳۲] PED-ML

دهه گذشته فیشینگ با سرقت اطلاعات حساس برای دستیابی به این امکانات، به تهدیدی جدی برای جامعه تبدیل شده است. این موضوع جزو سودآورترین جرایم اینترنتی محسوب می شود و طبق آمار محققان ای ام بی ایکس فورس[۳۳] انجام داده اند، تعداد افرادی که قربانی چنین فعالیت هایی می شوند به طرز چشمگیری در حال افزایش می باشد. از آنجا که خطر ایمیل های فیشینگ به طور مداوم در حال افزایش است، نیاز به شناسایی و غلبه بر چنین شرایطی یکی از مهمترین وظایف موجود است. در مقاله نام برده شده، نویسندگان از مدل غیر متوالی مانند رویکرد ماتریس اسناد اصطلاحی به دنبال تجزیه ارزش واحد[۳۴] و ضریب ماتریس غیر منفی[۳۵] برای ساخت مدل شناسایی ایمیل فیشینگ به عنوان یک مشکل طبقه بندی نظارت شده برای شناسایی ایمیل های فیشینگ از ایمیل های قانونی استفاده خواهند کرد.

(عبدل حاتم سابس و همکاران،۲۰۱۷)،سیستم هوشمند تشخیص وب سایت های فیشینگ با استفاده از دست بند جنگل تصادفی

فیشینگ به عنوان تقلیدی از وب سایت یک شرکت معتبر با هدف قرار دادن اطلاعات خصوصی کاربران آن شرکت تعریف می‌شود. به منظورپیش گیری از حملات فیشینگ، راه‌حل‌های مختلفی پیشنهاد شده‌است. با این حال، تنها یک گلوله جادویی نمی‌تواند این تهدید را به طور کامل از بین ببرد. داده‌کاوی یک تکنیک امیدوار کننده است که برای تشخیص حملات فیشینگ استفاده می‌شود. در مقاله نام برده شده، یک سیستم هوشمند برای تشخیص حملات فیشینگ توسط نویسندگان ارائه شده‌است. نویسندگان این مقاله از تکنیک‌های داده کاوی مختلف برای تصمیم‌گیری در مورد دسته‌های مختلف وب سایت‌ها استفاده کرده اند: قانونی یا فیشینگ.الگوریتم دسته بندی مختلفی به منظور ساخت سیستم های هوشمند با دقت بالا برای تشخیص وب سایت های فیشینگ مورد استفاده قرار گرفته اند وبرای ارزیابی دقت عملکرد تکنیک های داده کاوی از معیار ای یو سی[۳۶]

---

[۳۳] IBMs X-Force
[۳۴] Singular Value Decomposition (SVD)
[۳۵] Nonnegative Matrix Factorization (NMF)
[۳۶] AUC

که نشان دهنده مساخت زیر نمودار اراوسی[۳۷]است و اندازه گیری اف[۳۸]استفاده می شود.نتایج نشان بدست آماده توسط نویسندگان مقاله می دهند که جنگل تصادفی[۳۹]با دست یابی به دقت ۹۷,۳۶بهترین دقت وعمکرد را در بین دسته بند های مختلف برای تشخیص دارد و زمان اجرای جنگل تصادفی بسیار سریع است ومی تواند با وب سایت های مختلفی برای تشخیص فیشنگ برخورد کند.

(شاهن ماندال و همکاران ،-)،مروری برشناسایی لینک های فیشینگ با استفاده ازالگوریتم های خوشه بندی

فیشینگ نوعی حمله مهندسی اجتماعی می باشید.در این حملات مهاجم نقش یک نهاد قانونی را بازی میکند و از طریقی با قربانی راتباط برقرار میکند و از کاربر میخواهد لینکی را باز کند که از لحاظ ظاهری به گونه طراحی شده که شبیه به یک وب سایت قانونی است مهاجم اطلاعات کاربران را برای سرقت هویت و سرقت حساب ها و غیره ذخیزه میکند.در مقاله نام برده شده تمرکزنویسندگان بر روی حملات فیشینگ که مبتنی بر لینک هستند می باشد.راهکار های ارئه شده توسط نویسندگان این مقاله بیشتری بر روی الگوریتم های دسته بندی[۴۰]متمرکز شده است به نسبت الگوریتم های خوشه بندی[۴۱].هدف نویسندگان آزمایش و مقایسه نتایج هر دو نوع الگوریتم است تا متوجه شوند کدام یک عملکرد بهتری در شناسایی لینک های مخرب دارد.فرض اصلی رویکرد نویسندگان این مقاله یک مدل یادگیری ماشینی ترکیبی است که از دو مرحله :بررسی با لیست سیاه و لیست سفید و شناسایی مبتنی بر ابتکار برای افزایش دقت الگوریتم پیشنهاد شده توسط نویسندگان مقاله.

(نوشین امیری و همکاران، ۱۳۸۹)، استفاده از شبکه های عصبی خودرمزنگار موازی برای تشخیص صفحات جعلی اینترنتی

---

[۳۷] (ROC) curves
[۳۸] F-measure
[۳۹] Random Forest
[۴۰] Classification
[۴۱] Clustering

به کمک صفحات جعلی اینترنت تلاش می شود اطلاعات محرمانه یک کاربر مانند رمز حساب های بانکی و گذرواژه پست الکترونیکی به سرقت برده شود،این صفحات جعلی درواقع مشابه صفحات وب سایت های معتبر موجو مانند درگاه های پرداخت اینترنتی،یاهو و گوگل ساخته می شوندو به گونه ای کاربران به سمت این صفحات کشانده می شوند به این نوع حملات اینترنتی حمله فیشینگ گفته می شودتشخیص برخط صفحات فیشینگ به کمک نرم افزار های هوشمند میتواند از به سرقت رفتن اطلاعات کاربران جلوگیری کند و امنیت را در فضای وب افزاش دهد.در مقاله نام برده شده نویسندگان یک روش مبتنی بر شبکه های عصبی مصنوعی از نوع خود رمزنگارمعرفی کرده اند در روش پیشنهاد شده توسط محققین مقاله نام برده شده از دو شبکه عصبی خود رمزنگار موازی استفاده کرده اند که یکی از آنها با صفحات معمولی و ذیگیری با صفحات جعلی آموزش دیده است، در زمان تشخیص بر اساس بردار های رمز شده به دست آمده از هر دو شبکه موازی و یک لایه شبکه عصبی معمولی مانند سافت مکس و نوع صفحه ورودی را تشخیص می دهند، در کار برد های عملی هرگاه صفحه ورودی جعلی تشخیص داده شودبه سرعت از طریق مرور گر به کاربر اخطار داده می شود. یا دسترسی مسدود می شود. نتایج حاصل از آزمایش روش پیشنهادی به کمک مجموعه داده های فیشینگ وب سایت و معیار های صحت متوسط و دقت به خوبی نشان می دهند که شبکه های عصبی خود رمزنگار موازی عملکرد قوی تری نسبت به سایر روش های یادگیری ماشین در تشخیص صفحات جعلی اینترنتی دارد.

(مهدیه بهارلو و همکاران ، ۱۳۹۸)، بهبود روش شناسایی وب سایت فیشینگ با استفاده از داده‌کاوی روی صفحات وب

فیشینگ یک نوع حمله اینترنتی در سطح وب است که هدف آن سرقت مشخصات فردی کاربران برای دزدی آنلاین است. فیشینگ دارای اثر منفی در از بین بردن اعتماد بین کاربران در کسب‌وکارهای الکترونیکی است؛ بنابراین در مقاله نام برده شده سعی بر بررسی روشهای تشخیص وب سایت‌های فیشینگ با استفاده از داده کاوی شده است. شناسایی ویژگی‌های برجسته صفحات

فیشینگ یکی از پیش‌شرط‌های مهم در طراحی یک سیستم تشخیصی دقیق است؛ لذا در گام اول، برای شناسایی ویژگی‌های نفوذ فیشینگ یک لیست با ۳۰ ویژگی مطرح در وب‌سایت‌های فیشینگ آماده گردید. سپس برای افزایش کارایی سامانه‌های تشخیص فیشینگ روش جدیدی جهت کاهش ویژگی ها در دومرحله مبتنی بر انتخاب ویژگی و استخراج ویژگی پیشنهاد شده است که موجب می شود تعداد ویژگی‌ها به طور قابل توجهی کاهش یابند. پس‌ازآن عملکرد روش‌های درخت تصمیم جی ٤٨ ، جنگل تصادفی و بیزین ساده بر روی ویژگی‌های کاهش‌یافته مورد بررسی قرار گرفت شده است. نتایج حاصل از پژوهش محققان این مقاله نشان می‌دهد دقت مدل ایجاد شده برای تعیین صفحات فیشینگ با استفاده از کاهش ویژگی دومرحله‌ای مبتنی بر پوششی و الگوریتم تحلیل مؤلفه اصلی در روش جنگل تصادفی ٪۹٦،۵۸ می‌باشد که نسبت به سایر روش‌ها نتیجه مطلوبی است

(فاطمه صف آرا و همکاران ،۱۳۹۸)،افزایش دقت شناسایی صفحات جعلی وب با استفاده از الگوریم بهینه سازی کفتار و شبکه عصبی مصنوعی

ایجاد صفحات جعلی در محیط وب یا فیشینگ از جمله حملات سایبری است که نیازمند ملاحظات فرماندهی و کنترل می باشد.در حملات فیشینگ افراد به سمت صفحات جعلی که توسط فیشر یا سارق ساخته شده هدایت می شوند و اطلاعات مهم آنها توسط فیشربه سرقت می رود.الگوریتم های یادگیری ماشین و داده کاوی، الگوریتم های رایج برای طبقه بندی و تشخیص وبسایت های جعلی هستند.طبقه بندی وبسایت ها بر اساس ویژگی هایی که از آن سایت استخراج می شود صورت می گیرد. بنابراین انتخاب ویژگی تأثیر زیادی در نتایج طبقه بندی دارد. امروزه الگوریتم های فراابتکاری متعددی جهت انتخاب ویژگی و بهینه سازی عملکرد الگوریتم های طبقه بندی ارائه شده اند. در مقاله نام برده شده، الگوریتم فراابتکاری کفتار به منظور انتخاب ویژگی های مناسب برای طبقه بندی وبسایت های جعلی مورد استفاده قرار گرفته است. در این راستا، بهبودی بر الگوریتم فراابتکاری کفتار پیشنهاد شده والگوریتم کفتار بهبودیافته، ویژگی های مناسب را از

میان کل ویژگی های موجود انتخاب کرده و به شبکه عصبی مصنوعی ارسال می کند تا در جهت طبقه بندی وبسایت ها مورد استفاده قرار گیرند.نتایج پیاده سازی الگوریتم پیشنهادی توسط نویسندگان مقاله نام برده شده نشان می دهد که این الگوریتم با دقت نهایی۹۸/٦٤٪نسبت به الگوریتم استاندارد بهینه سازی کفتارعملکرد بهتری داشته است.علاوه بر این، نتایج حاکی از برتری نسبت به سه الگوریتم فراابتکاری بهینه سازی ذرات، کرم شب تاب و خفاش است. همچنین در مقاله نام برده شده ،الگوریتم پیشنهادی توسط محققین مقاله نامبرده شده با تعدادی از الگوریتم های طبقه بندی ارائه شده در پژوهش های پیشین روی مجموعه داده مشابه،مقایسه شده وبرتری آن نشان داده شده است.

(مهدی دادخواه و همکاران،۱۳۹۵)، ارائه رویکردی به منظور شناسایی و پیش بینی وب سایت های فیشینگ به وسیله الگوریتم های کلاس بندی بر اساس مشخصه های صفحات وب

امروزه مهمترین ریسک و چالش مورد توجه در تجارت و بانکدارای الکترونیک، خطر کلاهبرداری آنلاین و حملات فیشینگ است. حملات فیشینگ همواره به عنوان یکی از ابزارهای پرکاربرد برای مهاجمان، به منظور سرقت کلمه های عبور و رمزهای الکترونیک کاربران در مبادلات الکترونیک بوده است. در این نوع کلاهبرداری، مهاجمان نامه های الکترونیک با ادعاهای مختلف به قربانی ارسال می کند و با تکنیک های مختلفی قربانی را به صفحه های جعلی خود هدایت می کند سپس اقدام به سرقت اطلاعات حساس کاربران مانند رمزهای عبور می نماید. صفحات وب، نامه های الکترونیک و آدرسهای فیشینگ دارای ویژگی هایی هستند که از آن ها می توان برای شناسایی این حملات استفاده کرد. در مقاله نام برده شده رویکردی جهت شناسایی و پیش بینی وب سایت های فیشینگ با استفاده از الگوریتم های کلاس بندی بر اساس مشخصه های صفحات وب ارائه شده است که نرخ خطای کمتری نسبت به سایر تکنیک های مقابله با حملات فیشینگ، به خصوص تکنیک های مشابه مبتنی بر الگوریتم های داده کاوی دارد. در رویکرد ارائه شده توسط محققین مقاله نام برده شده، ویژگی های قابل استفاده در شناسایی صفحات فیشینگ بر اساس میزان تاثیر در

شناسایی این حملات وزن بندی شده سپس با اعمال الگوریتم های کلاس بندی بر روی مجموعه داده های مرتبط، الگویی به منظور شناسایی این حملات استخراج می گردد که قادر به شناسایی حملات فیشینگ بوده و نرخ خطای کمتری را نسبت به سایر روشهای مشابه پیشین نیز دارا می باشد.

(سلمان کمالی زاده و همکاران ،۱۳۹٤)، ارزیابی روش های شناسایی وب سایت فیشینگ

فیشینگ یکی از تکنیک‌های مهندسی اجتماعی برای فریب کاربران است که به معنای تلاش برای به دست آوردن اطلاعات محرمانه مانند نام کاربری، گذرواژه یا اطلاعات حساب بانکی است. امروزه از مهم‌ترین چالش‌های موجود در اینترنت، خطر حملات فیشینگ و کلاه‌برداری‌های اینترنتی است. این حملات تنها در آمریکا، سالیانه چندین میلیارد دلار خسارت به بار می‌آورد. از این‌رو، پژوهشگران تلاش‌های زیادی در جهت شناسایی و مقابله با این‌گونه حملات داشته‌اند. در مقاله نام برده شده هدف محققین، ارزیابی روش‌های شناسایی وب‌سایت های فیشینگ است. مقاله نام برده شده از نظر هدف کاربردی و از نظر ماهیت از نوع توصیفی – تحلیلی است. در مقاله نام برده شده، ضمن معرفی حمله فیشینگ و روش‌های موجود، شناسایی وب‌سایت فیشینگ، بر اساس مطالعات انجام شده و تجارب محققان مقاله نام برده شده با پیشنهاد معیارهایی، روش‌های شناسایی وب‌سایت فیشینگ مورد ارزیابی قرار داده اند. نتایج به دست آمده از پژوهش های محققین مقاله نامبرده شده حاکی از آن است که روش‌هایی که از تکنیک‌های مختلف شناسایی در کنار هم استفاده می‌کنند و همچنین اکثر ویژگی‌های صفحات وب را بررسی می‌کنند، در شناسایی حمله از موفقیت بیشتری برخوردار می‌باشند.

در جدول زیر روش های تحقیق مقالت مختلف را مورد بررسی و مقایسه قرار داده ایم:

نام مقاله	الگوریتم مورد استفاده	دقت	نرخ مثبت کاذب	روش
الگوریتم های فرایادگیری و درختان تصمیم اضافی برای شناسایی صفحات فیشینگ	آدابوست اکسترا تری ، بگینگ اکسترا تری، روتیشن فارست اکسترا تری، لجیت بوست اکسترا تری	۹۷.٤۸٥ دقت میانگین مدل ها	۰,۰۳۸	الگوریتم های نامبرده شده را روی داده ها آموزش داده و نتایج حاصل از آنها را با هم مقایسه کرده اند
ارزیابی روش های مبتنی برا یادگیری ماشین برای شناسایی سایت های فیشینگ	آدابوست، بگینگ ، ماشین‌های بردار پشتیبان ، درخت طبقه‌بندی و رگرسیون، لجستیک رگرسیون، جنگل‌های تصادفی، شبکه‌های عصبی، نایو بایز و درخت‌های رگرسیون افزایشی بیزی	۹۱،٦۲ دقت میانگین مدل ها	٦،۲۱	داده های خود را با استفاده از روش‌های تشخیص مبتنی بر یادگیری ماشین طبقه‌بندی می‌کنند، و عملکرد هر دسته بند را اندازه‌گیری می‌کنند. برای ارزیابی مدل های توسعه داده شده از معیار های اف وان ،نرخ خطا، و مساحت تحت منحنی اراو سی استفاده کرداند
دسته بندی لینک های فیشینگ با	شبکه های عصبی مصنوعی	۸٦.٦٤	۱۳.۹٦	شبکه های عصبی مختلفی را به همرا

استفاده از شبکه های عصبی عمیق رو دیتاست لینک های پر خطر				الگوریتم های مختلف روی داده ها آموزش داده و عملکذد آنها را با هم مقایسه کرده اند
شناسایی لینک های مخرب بر اساس باز ارسال در شبکه های اجتماعی آنلاین	بیز نت،جنگل تصادفی ،جی ٤٨	٨٤.٧٤ ٨٢.٢٢ ٧٩.٠٧	٩.٠٩ ١٠,٢٣ ١٠.٧٧	ایتدا از تکنیک های انتخاب ویژگی و استخراج ویژگی استفاده کرده اند سپس الگوریتم ها را آموزش داده و دقت و کارایی آنها را مورد ارزیابی قرار داده اند
سیستم هوشمند تشخیص وب سایت های فیشینگ با استفاده از دست بند جنگل تصادفی	جنگل تصادفی	٩٧,٣٦	_____	الگوریتم را با استفاده از داده ها آوزش داده سپس با معیار ای یوسی دقت آنرا مورد ارزیابی قرار داده اند و نتایج را روی نمودار ار او سی نمایش داده اند و الگوریتم های مختلف را با هم مقایسه کرده اند

یوارال نت :یادگیری لینک های نماینده با یادگیری عمیق برای شناسایی لینک های مخرب	شبکه های عصبی مصنوعی	۹۹,۲۹	۰.۷٦۸۳	شبکه های عصبی مختلف را طراحی و اموزش داده اند و کارایی هر مدام را بدست آورده اند
افزایش دقت شناسایی صفحات جعلی وب با استفاده از الگوریم بهینه سازی کفتار و شبکه عصبی مصنوعی	شبکه های عصبی ، الگوریتم بهینه سازی کفتار	۹۸,٦٤	_____	با استفاده از الگوریتم کفتار ویژگی های مناسب را به شبکه های عصبی داده و دقت آن را ارزیابی کرده اند.
استفاده از شبکه های عصبی خودرمزنگار موازی برای تشخیص صفحات جعلی اینترنتی	شبکه های عصبی خود رمزنگار	۹۳,٤	_____	از دو شبکه عصبی خود رمزنگار با معماری موازی استفاده کرده اند
بهبود روش شناسایی وب سایت فیشینگ با استفاده از داده‌کاوی روی صفحات وب	جنگل تصادفی ، بیزساده،جی ٤۷	۹۷,۲۵۹۲ ۹۲,۹۸۰۶ ۹۵,۹۷۴۷	_____	از روش های کاهش ابعاد و انتخاب ویژگی استفاده کرده سپس الگوریتم های خود را آموزش داده و دقت آنها را مورد ارزیابی قرار داده اند

فیشینگ چیست؟

فیشینگ نوعی حمله مهندسی اجتماعی است که اغلب برای سرقت اطلاعات کاربر از جمله شماره کارت اعتباری استفاده می شود.این نوع حملات زمانی اتفاق می‌افتند که یک مهاجم، خود را به عنوان یک موجودیت قابل اعتماد نشان داده و قربانی را فریب می‌دهد تا یک ایمیل، پیام فوری یا پیام متنی را باز کند. سپس گیرنده فریب داده می شود تا روی یک پیوند مخرب کلیک کند، که می تواند منجر به نصب بدافزار، مسدود شدن سیستم به عنوان بخشی از حمله باج افزار یا افشای اطلاعات حساس شود. حملات فیشینگ از روزهای اولیه اینترنت وجود داشته است. مجرمان سایبری اولین حملات فیشینگ را در اواسط دهه ۱۹۹۰ با استفاده از سرویس آمریکا آنلاین[۴۲] برای سرقت رمزهای عبور و اطلاعات کارت اعتباری مورد استفاده قرار دادند. در حالی که حملات مدرن از مدل های مهندسی اجتماعی مشابه ای استفاده می کنند، مجرمان سایبری از تاکتیک های تکامل یافته تری استفاده می کنند. فیشینگ در اصل یک روش حمله است که از تاکتیک‌های مهندسی اجتماعی استفاده می‌کند تا افراد را وادار به انجام اقدامی کند که بر خلاف منافع شان است. با درک بهتر انواع حملات فیشینگ و نحوه شناسایی آنها، سازمان ها می توانند به طور موثرتری از کاربران و داده های آنها محافظت کنند.انواع حملات فیشینگ عبارتند از:فیشینگ ایمیل[۴۳]،اچ تی تی پی اس فیشینگ[۴۴]،اسپیر فیشینگ[۴۵] ... می باشد.که هر کدام از این نوع حملات از روش های مختلفتی برای اسیب رساندن به کاربران استفاده می کنند و روش های مختلفی می توان آنها را شناسایی کرد

[۴۲] America Online(AOL)
[۴۳] Email phishing
[۴۴] HTTPS phishing
[۴۵] Spear phishing

در این پروژه تمرکز ما بر روی شناسایی وب سایت های فیشینگ و استخراج ویژگی مهم برای شناسایی صفحات فیشینگ می باشد.

ضرورت شناسایی حملات فیشینگ چیست؟

با توجه به روند رو به رشد اینترنت، تعداد رایانه های متصل به اینترنت روز به روز در حال افزایش است. تقریبا همه شرکت ها در حال انتقال اعمال اصلی خود از دنیای واقعی به دنیای سایبری هستند،اگر چه این موضوع باعث افزایش فروش آنها میشود میتواند آسیب پذیری های زیادی مانند حملات سایبری برای این شرکت ها را درپی داشته باشد به ویژه با ساختار ناشناس اینترنت.فیشینگ یکی از انواع محبوب حملات سایبری است که از ناآگاهی کاربر برای آسیب رساندن به آن ها سو استفاده می شود.فیشینگ یکی از شدیدترین حملات سایبری است که محققان علاقه مند به یافتن راه حلی برای آن هستند. در فیشینگ، مهاجمان کاربران نهایی را فریب می دهند و اطلاعات شخصی آنها را می دزدند. برای به حداقل رساندن آسیب ناشی از فیشینگ باید در اسرع وقت شناسایی شود. حملات فیشینگ مختلفی مانند سمیشینگ[۴۶]،ویشینگ[۴۷]... برای آسیب رساندن و سرقت اطلاعات کابران وجود دارد. تکنیک‌های مختلف تشخیص فیشینگ بر اساس لیست سفید[۴۸]،لیست سیاه[۴۹]،مبتنی بر محتوا[۵۰]،یادگیری ماشین و ... وجود دارد. در این پروژه تمرکز ما بر روی روش های مبتنی بر یادگیری ماشین می باشد. فیشینگ نوعی حمله مهندسی–وب اجتماعی[۵۱]در فضای مجازی است که در آن مجرمان اطلاعات ارزشمند کاربران ناآگاه را به سرقت می برند. کارآمد بودن اقدامات متقابل موجود در قالب نرم افزار ضد فیشینگ و روش های محاسباتی برای شناسایی فعالیت های فیشینگ اثبات شده است. با این حال، هکرها روش های جدیدی را برای خنثی سازی این اقدامات متقابل به کار می گیرند. با توجه به ماهیت تکامل یافته حملات فیشینگ، نیاز به اقدامات متقابل جدید و کارآمد بسیار احساس می شود زیرا تأثیر حملات فیشینگ اغلب فاجعه بار است. طرح

[۴۶] Šminshing
[۴۷] Vishing
[۴۸] whitelisting
[۴۹] blacklisting
[۵۰] content-based
[۵۱] social web-engineering

های هوش مصنوعی سنگ بنای روش های متُقابل مدرنی است که برای کاهش حملات فیشینگ استفاده می شود.

هوش مصنوعی[۵۲]هوش مصنوعی که گاهی اوقات هوش ماشینی نیز نامیده می شود شبیه سازی فرایندهای هوش طبیعی[۵۳]توسط ماشین ها به ویژه سیستم های رایانه ای است. به عبارت دیگر، هوش مصنوعی به سامانه‌هایی گفته می‌شود که می‌توانند واکنش‌هایی مشابه رفتارهای هوشمند انسانی از جمله، درک شرایط پیچیده، شبیه‌سازی فرایندهای تفکری و شیوه‌های استدلالی انسانی و پاسخ موفق به آن‌ها، یادگیری و توانایی کسب دانش و استدلال برای حل مسائل را داشته باشند،بطور خلاصه هوش مصنوعی را دانش ساخت وطراحی عامل هوشمند تعریف کرده اند.این علم کاربرد های فراوانی در علوم رایانه، علوم مهندسی،تجارت،پزشکی و بسیاری از علوم دیگر دارد بعنوان مثال: در پزشکی تجزیه و تحلیل صدا قلب،ربات های پرستار،ارائه مشاوره و پیش‌بینی احتمال مرگ بیمار برای هر روش جراحی....،در امورمالی و تجارت تجزیه وتحلیل بازار های مالی،پیش بینی قیمت سهام ها،معاملات الگوریتمی،مدیریت دارای و...از کاربرد های هوش مصنوعی در این علوم هستند. هوش مصنوعی، موضوعی بسیار گسترده است که شاخه‌های متعددی دارد. شاخه های هوش مصنوعی عبارتند از:یادگیری ماشینی[۵۴]،شبکه های عصبی[۵۵]سیستم های خبره[۵۶]، پردازش زبان طبیعی [۵۷]تشخیص گفتار[۵۸]و بینایی ماشین[۵۹] و- رباتیک [۶۰] و منطق فازی [۶۱]ست.

یادگیری عمیق[۶۲]یادگیری عمیق که در زبان فارسی به یادگیری ژرف نیز ترجمه شده است بخشی از خانواده یادگیری ماشینی می باشد که بر روش های تمرکز دارد که مبتنی بر الگوریتم های شبکه عصبی مصنوعی[۶۳]هستند. این الگوریتم ها د تلاش اند که مغز انسان را شبیه‌سازی کنند.به طور خلاصه در یادگیری عمیق شبکه های عصبی مصنوعی و الگوریتم های مشابه مغز بشر از مجموعه های عظیم داده مهارت های مورد نظر را فرا می

---

[۵۲] Artificial Intelligence
[۵۳] Natural Intelligence
[۵۴] machine learning
[۵۵] Neural network
[۵۶] Expert Systems
[۵۷] Natural language processing
[۵۸] speech recognition
[۵۹] Machine vision
[۶۰] robotic
[۶۱] Fuzzy logic
[۶۲] Deep learning
[۶۳] Artificial neural network

گیرند.همانطور که ما از طریق تجربه چیزهای جدید یاد می گیریم الگوریتم یادگیری عمیق نیز با هر بار تکرار یک کار مهارت خود را نسبت به دفعات قبلی بهبود می بخشد. دلیل استفاده از عبارت یادگیری عمیق این است که شبکه های عصبی لایه های مختلف یا عمیقی دارند که یادگیری را ممکن می سازد.

داده کاوی[۶۴]داده کاوی فرایندی برای تبدیل داده های خام به اطلاعات مفید می باشد،داده کاوی فرآیند استخراج و کشف الگوها در مجموعه داده های بزرگ است که شامل روش هایی در محل تلاقی یادگیری ماشین ، آمار و سیستم های پایگاه داده است.به عبارت دیگر داده کاوی یک زیرشاخه بین رشته ای علوم کامپیوتر و آمار با هدف کلی استخراج اطلاعات (با روشهای هوشمند) از مجموعه داده و تبدیل اطلاعات به یک ساختار قابل درک برای استفاده بیشتر است.

یادگیری ماشین چیست؟

یادگیری ماشینی شاخه‌ای از هوش مصنوعی[۶۵]و علوم کامپیوتر[۶۶]است که بر استفاده از داده‌ها و الگوریتم‌ها برای تقلید از روشی که انسان‌ها یاد می‌گیرند تمرکز دارد و به تدریج دقت آن را بهبود می‌بخشد. یادگیری ماشین به عنوان بخشی از هوش مصنوعی در نظر گرفته می شود که به مطالعه الگوریتم های کامپیوتری می پردازد که می تواند به طور خودکار از طریق تجربه و با استفاده از داده ها بهبود یابد. الگوریتم‌های یادگیری ماشین مدلی را بر اساس داده‌های نمونه می‌سازند که به داده‌های آموزشی معروف است تا پیش‌بینی‌ها یا تصمیم‌گیری‌ها را بدون برنامه‌ریزی صریح انجام دهند.الگوریتم‌های یادگیری ماشین در کاربردهای متنوعی مانند پزشکی، فیلتر کردن ایمیل، تشخیص گفتار و بینایی کامپیوتری استفاده می‌شوند.انواع الگوریتم های یادگیری ماشین عبارت اند از:

[۶۴] Data mining
[۶۵] artificial intelligence
[۶۶] computer science

درخت تصمیم[۶۵]،لاجستیک رگرسیون[۶۶]،ماشین بردار پشتیبان[۶۷]،دسته بند بیز[۶۸]،نزدیک ترین همسایه[۶۹]....برخی از الگوریتم های استفاده شده در این پژوهش به شرح زیر می باشند:

درخت تصمیم:

الگوریتم درخت تصمیم به خانواده الگوریتم های یادگیری ماشین با نظارت[۷۰] تعلق دارد. می توان از این الگوریتم برای حل مسائل طبقه بندی و رگرسیون استفاده کرد. هدف این الگوریتم ایجاد مدلی است که مقدار یک متغیر هدف را پیش بینی می کند، که برای این منظور درخت تصمیم از نمایش درختی برای حل مسئله استفاده می کند. گره برگ مربوط به یک برچسب کلاس است و ویژگی ها در گره داخلی درخت نشان داده می شوند.

جنگل تصادفی:[۷۱]

جنگل‌های تصادفی یک روش یادگیری جمعی[۷۲] برای طبقه‌بندی، رگرسیون و سایر وظایف است که با ساختن تعداد زیادی درخت تصمیم در زمان آموزش عمل می‌کند. برای کارهای طبقه بندی، خروجی جنگل تصادفی کلاسی است که توسط اکثر درختان انتخاب شده است. برای وظایف رگرسیون، میانگین یا میانگین پیش‌بینی درختان منفرد برگردانده می‌شود.

نایو بیز:[۷۳]

الگوریتم ساده بیز یک الگوریتم یادگیری نظارت شده است که بر اساس قضیه بیز است و برای حل مسائل طبقه بندی استفاده می شود.طبقه‌بندی‌کننده ساده بیز یکی از ساده‌ترین و مؤثرترین

---

[۶۵] Decision tree
[۶۶] Logistic Regression
[۶۷] Support vector machine
[۶۸] Naive Bayes Classifiers
[۶۹] KNN
[۷۰] supervised machine learning
[۷۱] Random Forest
[۷۲] ensemble learning
[۷۳] Naïve Bayes Classifier

الگوریتم‌های طبقه‌بندی است که به ساخت مدل‌های یادگیری ماشین سریع کمک می‌کند که بتوانند پیش‌بینی‌های سریع انجام دهند.

انتخاب ویژگی چیست؟

هنگام ساخت یک مدل یادگیری ماشین در زندگی واقعی، تقریباً نادر است که همه متغیرهای مجموعه داده برای ساخت یک مدل مفید باشند .افزودن متغیرهای اضافی قابلیت تعمیم مدل را کاهش می دهد و همچنین ممکن است دقت کلی یک طبقه بندی کننده را کاهش دهد .علاوه بر این، افزودن متغیرهای بیشتر و بیشتر به یک مدل، پیچیدگی کلی مدل را افزایش می‌دهد.انتخاب ویژگی فرآیند کاهش تعداد متغیرهای ورودی هنگام توسعه یک مدل پیش‌بینی کننده است. کاهش تعداد متغیرهای ورودی برای کاهش هزینه محاسباتی مدل سازی و در برخی موارد برای بهبود عملکرد مدل مطلوب است. روش‌های انتخاب ویژگی مبتنی بر آمار شامل ارزیابی رابطه بین هر متغیر ورودی و متغیر هدف با استفاده از آمار و انتخاب آن دسته از متغیرهای ورودی است که قوی‌ترین رابطه را با متغیر هدف دارند. این روش ها می توانند سریع و موثر باشند، اگرچه انتخاب معیارهای آماری به نوع داده متغیرهای ورودی و خروجی بستگی دارد. هدف از انتخاب ویژگی در یادگیری ماشینی یافتن بهترین مجموعه از ویژگی‌ها است که به فرد امکان می‌دهد مدل‌های مفیدی از پدیده‌های مورد مطالعه بسازد.تکنیک های انتخاب ویژگی در یادگیری ماشینی را می توان به طور کلی به دسته های زیر طبقه بندی کرد:

نظارت شده؛[۷۶]

این تکنیک ها را می توان برای داده های برچسب دار استفاده کرد و برای شناسایی ویژگی های مرتبط برای افزایش کارایی مدل های نظارت شده مانند طبقه بندی و رگرسیون استفاده می شود.

[۷۶] supervised

بدون نظارت:[۷۷]

این تکنیک ها را می توان برای داده های بدون برچسب استفاده کرد.

از آنجا که مسئله ما یک مسئله طبقه بندی می باشد به شرح چند مورد از روش ها انتخاب ویژگی مرتبط با طبقه بندی می پردازیم:

۱- فیلتر کردن:[۷۸]

روش‌های فیلتر، ویژگی‌های ذاتی ویژگی‌های اندازه‌گیری شده را از طریق آمار تک متغیره به جای عملکرد اعتبارسنجی متقابل، انتخاب می‌کنند. این روش‌ها سریع‌تر و از نظر محاسباتی هزینه کمتری نسبت به روش‌های وارپپر[۷۹]دارند. هنگام برخورد با داده های با ابعاد بالا، از نظر محاسباتی استفاده از روش های فیلتر ارزان تر است.

۲-وارپپر:

وارپپر ها به روشی برای جستجوی فضای همه زیرمجموعه های ممکن ویژگی ها، ارزیابی کیفیت آنها با یادگیری و ارزیابی طبقه بندی کننده با آن زیر مجموعه ویژگی نیاز دارند. فرآیند انتخاب ویژگی بر اساس یک الگوریتم یادگیری ماشین خاص است که ما سعی می کنیم آن را بر روی یک مجموعه داده معین قرار دهیم. این یک رویکرد جستجوی حریصانه را با ارزیابی همه ترکیب‌های ممکن از ویژگی‌ها در برابر معیار ارزیابی دنبال می‌کند. روش‌های وارپپرمعمولاً منجر به دقت پیش‌بینی بهتری نسبت به روش‌های فیلتر می‌شوند. ایده انتخاب زیرمجموعه ویژگی ها این است که بتوانیم بهترین ویژگی هایی را که برای کار طبقه بندی مناسب است پیدا کنیم. ما باید درک کنیم

---

[۷۷] unsupervised
[۷۸] Filter
[۷۹] wrapper

که همه ویژگی‌ها تاثیر یکسانی روی خروجی مدل نداشته و برخی ممکن است مرتبط‌تر از بقیه باشند.

الگوریتم بهینه ساز ازدحام ذارت:

در علوم محاسباتی، بهینه‌سازی ازدحام ذرات یک روش محاسباتی است که با تلاش مکرر برای بهبود راه‌حل کاندید با توجه به معیاری از کیفیت، یک مسئله را بهینه می‌کند. با داشتن جمعیتی از راه‌حل‌های کاندید، که در اینجا ذرات نامیده می‌شوند، و حرکت دادن این ذرات در فضای جستجو بر اساس فرمول ساده ریاضی بر روی موقعیت و سرعت ذره، مشکلی را حل می‌کند. بهینه‌سازی ازدحام ذرات یکی از الگوریتم‌های الهام‌گرفته ازطبیعت است و روش کار آن بسیار ساده است به این صورت که راه‌حل بهینه را در فضای راه‌حل جستجو می کند. با سایر الگوریتم های بهینه سازی متفاوت است به گونه ای که فقط تابع هدف مورد نیاز است و وابسته به گرادیان یا هر شکل دیفرانسیل هدف نیست. همچنین دارای هایپرپارامترهای بسیار کمی است. بهینه‌سازی ازدحام ذرات ابتدا توسط کندی و ابرهارت به‌عنوان یک الگوریتم تکاملی مبتنی بر جمعیت برای شبیه‌سازی رفتار مشارکتی پرندگان در یافتن غذا ابداع شد. مزیت اصلی این الگوریتم نسبت به سایر الگوریتم‌های بهینه‌سازی، توانایی آن در دستیابی به همگرایی سریع در بسیاری از مسائل بهینه‌سازی پیچیده است.علاوه بر این، این الگوریتم دارای چندین مزیت جذاب از جمله سادگی با معادلات ریاضی کمتر و داشتن پارامترهای کمتر در پیاده سازی است.

ابزار ها:

با توجه به اینکه با پایتون به راحتی می‌توان فرآیندهای دشوار را مدیریت کرد و استفاده از آن ساده است، در این پروژهش ما از زبان برنامه نویسی پایتون برای توسعه مدل ها خود استفاده میکنیم همچنین این زبان برنامه نویسی کتابخانه های فراوانی برای کار با الگورتیم های هوش مصنوعی و پردازش داده ها را دارا می باشد. پایتون مجموعه وسیعی از کتابخانه‌ها را برای توسعه هوش

مصنوعی ارائه می‌دهد که شامل موارد پایه‌ای نیز هست که در زمان برنامه نویسی، صرفه جویی می‌کند. پایتون به دلیل کد جمع و جور و خواندنی‌اش مشهور است و از نظر قابلیت استفاده عملاً بی‌نظیر است. ساده و مختصر بودن پایتون دلیلی‌ست که آن را با سایرزبان‌های برنامه نویسی متفاوت می‌کند و به زمان کدنویسی کمتری نیاز دارد. همچنین به توسعه دهنده اجازه می‌دهد تا الگوریتم‌ها را بدون اجرا کردن، سریع آزمایش کند.

داده ها:

به منظور آموزش واعتبار سنجی روش پیشنهادی از مجموعه داده های فیشینگ وب سایت پایگاه داده یو سی ای‌آی[۸]استفاده شده است این مجموعه داده شامل ۱۱۰۵۵ نمونه وب سایت است که۶۱۵۷وب سایت از آنها فیشینگ و۴۸۹۸ مورد دیگر وب سایت های مشروع هستند، از هر وب سایت ۳۰ ویژگی استخراج شده استاین ویژگی ها عمدتا صحیح و دومقداری هستند که به شرح زیر می باشند:

[۸] UCI

دسته	ویژگی	مقدار
ویژگی‌های مرتبط با آدرس سایت	Using the IP Address	−1, 1
	Long URL to Hide the Suspicious Part	1, 0, −1
	Using URL Shortening Services “TinyURL”	1, −1
	URL’s having “@” Symbol	1, −1
	Redirecting using “//”	−1, 1
	Adding Prefix or Suffix Separated by (-) to the Domain	−1, 1
	Sub Domain and Multi Sub Domains	−1, 0, 1
	HTTPS (Hyper Text Transfer Protocol with Secure Sockets Layer)	−1, 1, 0
	Domain Registration Length	−1, 1
	Favicon	1, −1
	Using Non-Standard Port	1, −1
	The Existence of “HTTPS” Token in the Domain Part of the URL	−1, 1
ویژگی‌های غیرعادی مرتبط با صفحات و پیوندها	Request URL	1, −1
	URL of Anchor	−1, 0, 1
	Links in <Meta>, <Script> and <Link> tags	1, −1, 0
	Server Form Handler (SFH)	−1, 1, 0
	Submitting Information to Email	−1, 1
	Abnormal URL	−1, 1
ویژگی‌های مرتبط با اطلاعات کد منبع سایت	Website Forwarding	0, 1
	Status Bar Customization	1, −1
	Disabling Right Click	1, −1
	Using Pop-up Window	1, −1
	IFrame Redirection	1, −1
ویژگی‌های مرتبط با دامنه	Age of Domain	−1, 1
	DNS Record	−1, 1
	Website Traffic	−1, 0, 1
	PageRank	−1, 1
	Google Index	1, −1
	Number of Links Pointing to Page	1, 0, −1
	Statistical-Reports Based Feature	−1, 1

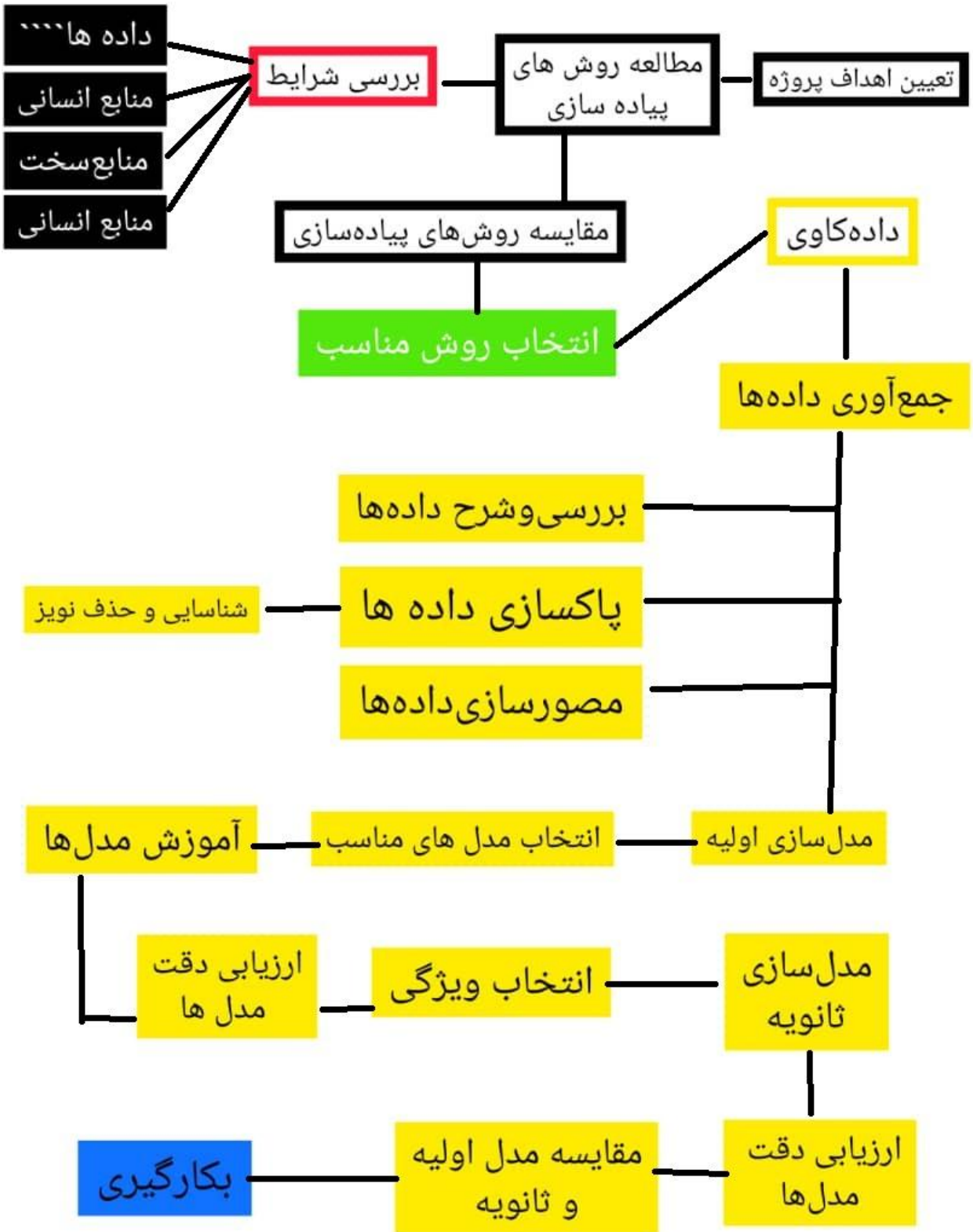
روش کار در این پژوهش:

اگر مجموعه ای از ویژگی ها به ما داده شود، چگونه می توانیم بهینه ترین زیر مجموعه را بشناسیم؟ از آنجا که ما میخواهیم انتخاب ویژگی ها برای هر الگوریتم دسته بندی به صورت انحصاری صورت گیرد در این پروژه از روش دوم یعنی وارپپر برای انتخاب ویژگی استفاده میکنیم.تا اینکار

باعث افزایش دقت مدل ها شود.برای وزن دهی به ویژگی ها ما از الگوریتم بهینه ساز ازدحام ذرات[۸۱] استفاده میکنیم،روش کار این الگوریتم به این صورت می باشد که برای هر مجموعه داده یک تابع هزینه،هزینه را محاسبه می کند و سپس الگوریتم ازدحام ذرات سعی می کند که مجموعه داده ای را پیدا کند که برای الگوریتم دسته بند مورد نظر کمترین تابع هزینه را به همراه داشته باشد بدین منظور ما یک تابع هزینه نیز نیاز داریم که الگوریتم بهینه ساز ازدحام ذرات آن را برای هر الگوریتم طبقه بندی بهنیه کند. در این پژوهش علاوه بر شناسایی صفحات فیشینگ با الگوریتم های یادگیری ماشین ما می خواهیم با استفاده از روش های انتخاب ویژگی هزینه آموزش را کاهش و کیفیت مدل ها را افزایش داده برای این منظور ابتدا مدل ها انتخاب شده را با تمام داده ها آموزش داده و دقت آنها را سنجیده سپس با استفاده از روش های انتخاب ویژگی[۸۲] ویژگی های را که بیشترین تاثیر را روی خروجی مدل داشته شناسایی میکنیم و الگوریتم ماشین بردار پشتیبان با کرنل ار بی اف را با کمک الگوریتم ازدحام ذرات بهینه میکنیم و دقت مدل توسعه داده شده را محاسبه کرده و با سایر الگوریتم ها مقایسه میکنیم.

---

۸۱ particle swarm optimization
۸۲ Feature selection


داده ها
منابع انسانی
منابع‌سخت
منابع انسانی
بررسی شرایط
مطالعه روش های پیاده سازی
تعیین اهداف پروژه
مقایسه روش‌های پیاده‌سازی
انتخاب روش مناسب
داده‌کاوی
جمع‌آوری داده‌ها
بررسی‌وشرح داده‌ها
پاکسازی داده ها
شناسایی و حذف نویز
مصورسازی‌داده‌ها
مدل‌سازی اولیه
انتخاب مدل های مناسب
آموزش مدل‌ها
ارزیابی دقت مدل ها
انتخاب ویژگی
مدل‌سازی ثانویه
ارزیابی دقت مدل‌ها
مقایسه مدل اولیه و ثانویه
بکارگیری

شرح پروژه:

ابتدا کتابخونه های پانداز:برای خوندن فایل و تجزیه و تحلیل دیتافریم،نامپای:برای کار با آرایه های نامپای،مت پلات لیب:برای رسم نمودار،سیبورن:برای رسم نمودار،سایکیت لرن برای:استفاده از الگوریتم های یادگیری ماشینی،استاندارد کردن دیتا و اندازه گیری خطا فراخوانی شده اند.همچنین کتابخانه پایسوارم برای کار با الگوریتم ازدحام ذرات فراخوانی شده است.دیتاست مورد نظر ما با فرمت ای اراف اف[۸۳]ذخیره شده است که این فرمت مناسب کار در محیط پایتون نمی باشد به همین علت با قطعه کد زیر ما فرمت فایل را به سی اس وی[۸۴]تغییر می دهیم:

[۸۳].arff
[۸۴]ċsv

```
[11] path_to_directory="/content/"
files = [arff for arff in os.listdir(path_to_directory) if arff.endswith(".arff")]

def toCsv(content):
    data = False
    header = ""
    newContent = []
    for line in content:
        if not data:
            if "@attribute" in line:
                attri = line.split()
                columnName = attri[attri.index("@attribute")+1]
                header = header + columnName + ","
            elif "@data" in line:
                data = True
                header = header[:-1]
                header += '\n'
                newContent.append(header)
        else:
            newContent.append(line)
    return newContent


for z,file in enumerate(files):
    with open(path_to_directory+'/'+file , "r") as inFile:
        content = inFile.readlines()
        name,ext = os.path.splitext(inFile.name)
        new = toCsv(content)
        with open(name+".csv", "w") as outFile:
            outFile.writelines(new)
```

مجموعه داده ما داری ۳۱ ستون و۱۱۰۵۵سطر می باشد، تمام ستون ها شامل مقادیری کتگوریکال می باشند و در این مجموعه داده هیچ سطری با مقادی نامعلوم[۸۵]وجود ندارد.قطعه کد زیر پنج سطر ابتدای مجموعه داده ما را نشان می دهد:

```
[15] train.head()
```

	having_IP_Address	URL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Domain_registeration_length	Favicon	port	HTTPS_token	Request_URL	URL_of_Anchor	Links_in_tags	SFH
0	-1	1	1	1	-1	-1	-1	-1	-1	1	1	-1	1	-1	1	-1
1	1	1	1	1	1	-1	0	1	-1	1	1	-1	1	0	-1	-1
2	1	0	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	-1	-1
3	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	0	-1
4	1	0	-1	1	1	-1	1	1	-1	1	1	1	1	0	0	-1

کدهای زیر توصیفی از ویژگی ها را به صورت دیتافریم نمایش می دهد:

[۸۵] NAN

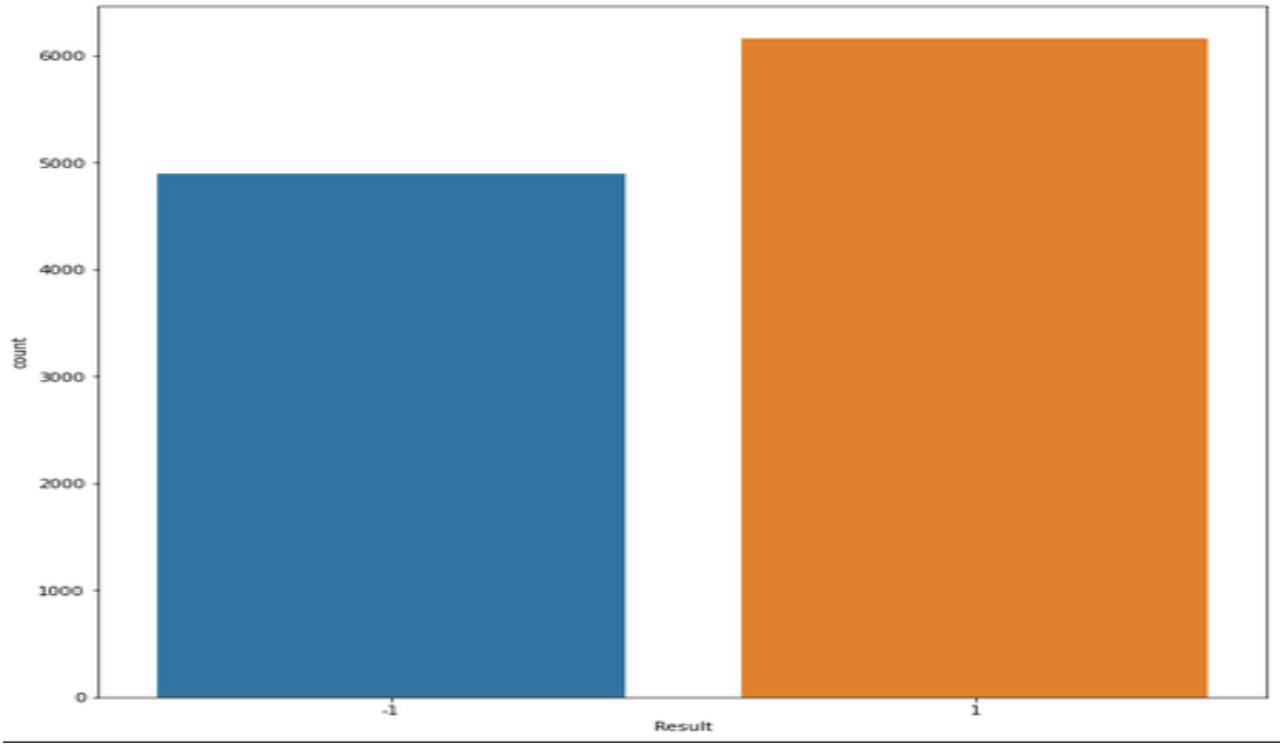
```
train.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
having_IP_Address	11055.0	0.313795	0.949534	-1.0	-1.0	1.0	1.0	1.0
URL_Length	11055.0	-0.633198	0.766095	-1.0	-1.0	-1.0	-1.0	1.0
Shortining_Service	11055.0	0.738761	0.673998	-1.0	1.0	1.0	1.0	1.0
having_At_Symbol	11055.0	0.700588	0.713598	-1.0	1.0	1.0	1.0	1.0
double_slash_redirecting	11055.0	0.741474	0.671011	-1.0	1.0	1.0	1.0	1.0
Prefix_Suffix	11055.0	-0.734962	0.678139	-1.0	-1.0	-1.0	-1.0	1.0
having_Sub_Domain	11055.0	0.063953	0.817518	-1.0	-1.0	0.0	1.0	1.0
SSLfinal_State	11055.0	0.250927	0.911892	-1.0	-1.0	1.0	1.0	1.0
Domain_registeration_length	11055.0	-0.336771	0.941629	-1.0	-1.0	-1.0	1.0	1.0
Favicon	11055.0	0.628584	0.777777	-1.0	1.0	1.0	1.0	1.0
port	11055.0	0.728268	0.685324	-1.0	1.0	1.0	1.0	1.0
HTTPS_token	11055.0	0.675079	0.737779	-1.0	1.0	1.0	1.0	1.0
Request_URL	11055.0	0.186793	0.982444	-1.0	-1.0	1.0	1.0	1.0
URL_of_Anchor	11055.0	-0.076526	0.715138	-1.0	-1.0	0.0	0.0	1.0
Links_in_tags	11055.0	-0.118137	0.763973	-1.0	-1.0	0.0	0.0	1.0
SFH	11055.0	-0.595749	0.759143	-1.0	-1.0	-1.0	-1.0	1.0
Submitting_to_email	11055.0	0.635640	0.772021	-1.0	1.0	1.0	1.0	1.0
Abnormal_URL	11055.0	0.705292	0.708949	-1.0	1.0	1.0	1.0	1.0
Redirect	11055.0	0.115694	0.319872	0.0	0.0	0.0	0.0	1.0
on_mouseover	11055.0	0.762099	0.647490	-1.0	1.0	1.0	1.0	1.0
RightClick	11055.0	0.913885	0.405991	-1.0	1.0	1.0	1.0	1.0
popUpWidnow	11055.0	0.613388	0.789818	-1.0	1.0	1.0	1.0	1.0
Iframe	11055.0	0.816915	0.576784	-1.0	1.0	1.0	1.0	1.0
age_of_domain	11055.0	0.061239	0.998168	-1.0	-1.0	1.0	1.0	1.0
DNSRecord	11055.0	0.377114	0.926209	-1.0	-1.0	1.0	1.0	1.0
web_traffic	11055.0	0.287291	0.827733	-1.0	0.0	1.0	1.0	1.0
Page_Rank	11055.0	-0.483673	0.875289	-1.0	-1.0	-1.0	1.0	1.0
Google_Index	11055.0	0.721574	0.692369	-1.0	1.0	1.0	1.0	1.0
Links_pointing_to_page	11055.0	0.344007	0.569944	-1.0	0.0	0.0	1.0	1.0
Statistical_report	11055.0	0.719584	0.694437	-1.0	1.0	1.0	1.0	1.0
Result	11055.0	0.113885	0.993539	-1.0	-1.0	1.0	1.0	1.0

قطعه کد زیر نمودار همبستگی بین ویژگی ها را رسم و با فرمت پی ان جی ذخیره می کند:

```
plt.figure(figsize=(30,40),dpi=100)
sb.heatmap(train.corr(),annot=True,linewidths=0.2)
plt.title("Relationship between features and label")
plt.svaefig("/content/corr_n.png")
```

نمودار زیر نشان میدهد که چه میزان از لینک هایموجود در دیتاست مخرب و چه میران از لینک ها سالم هستند:



قطعه کد زیر میزان داده های متعلق به هر مقدار عدد در هر ستون را نشان میدهد:

```
for column in train.columns:
    print("value count of "+column+":" '\n',train[column].value_counts())
    print(20*"=+=")
```

در عکس زیر مقادیر یکتای موجود در هر ستون را می بینیم:

```
for column in train.columns:
    print(column+":",train[column].unique())
```

```
having_IP_Address: [-1  1]
URL_Length: [ 1  0 -1]
Shortining_Service: [ 1 -1]
having_At_Symbol: [ 1 -1]
double_slash_redirecting: [-1  1]
Prefix_Suffix: [-1  1]
having_Sub_Domain: [-1  0  1]
SSLfinal_State: [-1  1  0]
Domain_registeration_length: [-1  1]
Favicon: [ 1 -1]
port: [ 1 -1]
HTTPS_token: [-1  1]
Request_URL: [ 1 -1]
URL_of_Anchor: [-1  0  1]
Links_in_tags: [ 1 -1  0]
SFH: [-1  1  0]
Submitting_to_email: [-1  1]
Abnormal_URL: [-1  1]
Redirect: [0 1]
on_mouseover: [ 1 -1]
RightClick: [ 1 -1]
popUpWidnow: [ 1 -1]
Iframe: [ 1 -1]
age_of_domain: [-1  1]
DNSRecord: [-1  1]
web_traffic: [-1  0  1]
Page_Rank: [-1  1]
Google_Index: [ 1 -1]
Links_pointing_to_page: [ 1  0 -1]
Statistical_report: [-1  1]
Result: [-1  1]
```

با استفاده از کد زیر داده ها برای آموزش و اعتبارسنجی مدل ها به دو بخش ترین و تست تقسیم می کنیم:

```
[29] X=train.drop("Result",axis=1).values
     y=train.Result
     x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42,)
```

قطعه کد زیر تمام مدل ها را می سازد:

```
[30] ##models
dt_gini=DecisionTreeClassifier(criterion="entropy",max_depth=5)
dt_entropy=DecisionTreeClassifier(criterion="gini",max_depth=5)
knn=KNeighborsClassifier(n_neighbors=5)

rf=RandomForestClassifier()

lr=LogisticRegression(random_state=42)

mlp=MLPClassifier(hidden_layer_sizes=(40,),max_iter=500)

poly_svm=SVC(kernel='poly')
linear_svm=SVC(kernel='linear')
sigmoid_svm=SVC(kernel='sigmoid')
rbf_svm=SVC(kernel='rbf')

model_list=[dt_gini,dt_entropy,knn,rf,lr,mlp,poly_svm,linear_svm,sigmoid_svm,rbf_svm]
```

قطعه کد زیر مدل ها را با دیتا ها که از قبل آماده شده اند آموزش داده و سپس دقت آنها را با معیار های مختلفی می سنجد و نمودار های کانفیژن ماتریکس را برای هر الگوریتم رسم می کند:

```
train_list=list()
test_list=list()
for model in model_list:
  model.fit(X,y)
  train_score=model.score(x_train,y_train)
  test_score=model.score(x_test,y_test)
  print("train score: ",train_score,"and test score is: ",test_score)
  test_list.append(test_score)
  train_list.append(train_score)
  predict_test=model.predict(x_test)
  print('\n',classification_report(y_test,predict_test),'\n')
  cm=confusion_matrix(y_test,predict_test)
  ax= plt.subplot()
  sb.heatmap(cm, annot=True, fmt='g', ax=ax);
  ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
  ax.set_title('Confusion Matrix');
  plt.show()
  print('\n'+10*"*_* "+'\n')
```

تابع لاس پر پارتیکال[۴۶] که در زیر به نمایش گذاشته شده است مقدار تابع هزینه را برای هر ذره محاسبه کرده این تابع توسط تابع لاس[۴۷] فراخوانی میشود که خود تابع لاسمجموعه لاس ها محاسبه شده توسط لاس پر پارتیکال را به صورت یک آرایه نامپای برمیگرند:

```
def loss_per_particle(m, alpha):

    total_features = 30
    if np.count_nonzero(m) == 0:
        X_sub = X.value
    else:
        X_sub = X[:,m==1]

    classifier.fit(X_sub, y)

    P = (classifier.predict(X_sub) == y).mean()

    j = (alpha * (1.0 - P)+ (1.0 - alpha) * (1 - (X_sub.shape[1] / total_features)))
    return j
```

```
def loss(x, alpha=0.88):

    n_particles = x.shape[0]
    j = [loss_per_particle(x[i], alpha) for i in range(n_particles)]
    return np.array(j)
```

در ادامه ما بهینه ساز ازدحام ذرات را تعریف کرده ایم با کمک یک حلقه و دو تابعی که در بالا نام برده شده است سعی کردیم برای هر الگوریتم بهترین مجموعه داده را پیدا کنیم و با استفاده از کد زیر هر داده روی مجموعه داده انحصاری خود آموزش داده شده و سپس دقت آن با معیار های مختلفی سنجیده شده است و در پایان دیتافریمی ایجاد شده است که دقت هر الگوریتم در شرایط مختلف نشان میدهد:

[۴۶] loss_per_particle
[۴۷] loss

pso

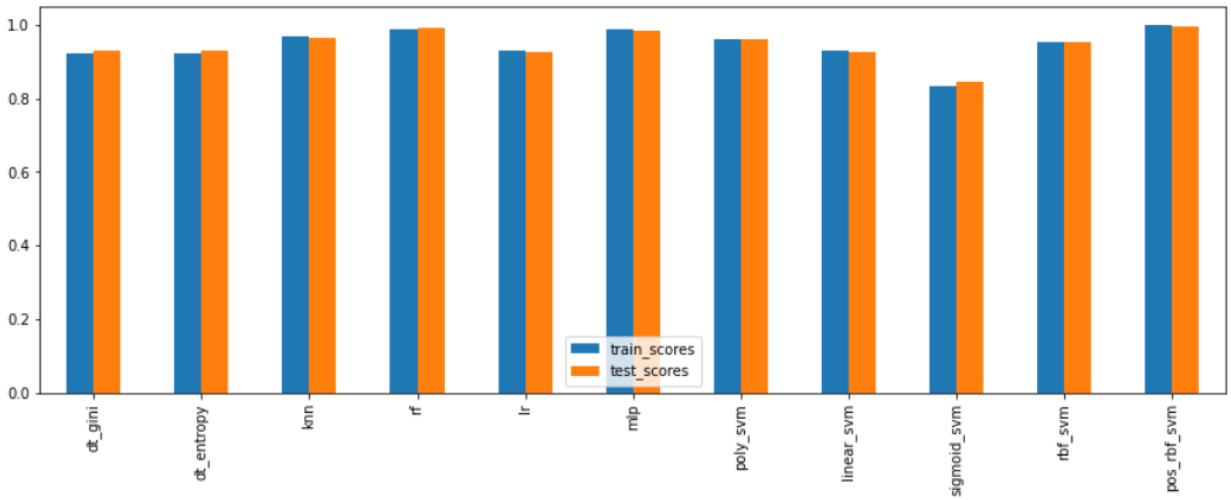
```
In [ ]: class PSO(object):
            def __init__(self,particle_num,particle_dim,iter_num,c1,c2,w,max_value,min_value):

                self.particle_num = particle_num
                self.particle_dim = particle_dim
                self.iter_num = iter_num
                self.c1 = c1
                self.c2 = c2
                self.w = w
                self.max_value = max_value
                self.min_value = min_value



            def swarm_origin(self):
                particle_loc = []
                particle_dir = []
                for i in range(self.particle_num):
                    tmp1 = []
                    tmp2 = []
                    for j in range(self.particle_dim):
                        a = random.random()
                        b = random.random()
                        tmp1.append(a * (self.max_value - self.min_value) + self.min_value)
                        tmp2.append(b)
                    particle_loc.append(tmp1)
                    particle_dir.append(tmp2)

                return particle_loc,particle_dir

            def fitness(self,particle_loc):

                fitness_value = []
                for i in range(self.particle_num):
                    rbf_svm = SVC(kernel = 'rbf', C = particle_loc[i][0], gamma = particle_loc[i][1])
                    cv_scores = cross_val_score(rbf_svm,trainX,trainY,cv =3,scoring = 'accuracy')
```

```
[ ] score_df["train_score_pos"]=pos_train_list
    score_df["test_score_pos"]=pos_test_list
    score_df
```

تذکر:این بخش از کد به دلیل سنگین بودن محاسبات آن در زمان اجرا نیازمند سخت افزار قوی برای اجرا می باشد ،متاسفانه به همین علت خروجی این قطعه کد برای بنده در دسترس نمی باشد!!!

نتایج :

تعداد حملات فیشینگ نسبت به گذشته رشد چشم گیری داشته است جلوگیری و شناسایی به موقع این حملات تشخیص صفحات فیشینگ علاوه بر اینکه امنیت کاربران را افزایش دهد می تواند از وارد شدن لطمه به اعتبار کسب و کار ها جلوگیری کند و از ضرر و زیان های بسیاری جلوگیری کند. یکی از راهکارای پیشنهاد شده برای شناسایی حملات فیشینگ استفاده از هوش مصنوعی برای تشخیص صفحات فیشینگ می باشد، الگوریتم های هوش مصنوعی می توانند داده های بسیار زیادی را در لحضه تحلیل کرد و به سرعت نتیجه را ارائه دهند. به همین علت هوش مصنوعی میتواند سرعت سیستم های تشخیص صفحات فیشینگ را افزایش دهد.ما در این پژوهش تلاش کردیم تا الگوریتم های مختلف هوش مصنوعی را جهت شناسایی صفحات فیشینگ کورد ارزیابی قرار دهیم که این الگوریتم ها عمکلرد خوبی را داشته اند ما میتوانیم به کمک تکنیک های انتخاب ویژگی ،ویژگی های که در شناسایی صفحات فیشینگ نقش بسزایی دراند را شناسایی کرده و الگوریتم ها را فقط با این ویژگی های استخراج شده آموزش دهیم این کار سرعت آموزش الگوریتم ها را افزایش داده وهمچنین سرعت شناسایی آنها توسط مدل ها را افزایش میدهد ،الگوریتم ازدحام ذرات میتواند ویژگی های مهم متناسب با هر الگوریتم طبقه بندی را شناسایی کند و سبب افزاریش دقت آنها شود ، نتایج حاصل از مقایسه نشان می دهد که الگوریتم ها داری سرعت بالاتری نسبت به حالت اولیه هستن و در زمان آموزش آنها کاهش میابد و از دادن اطلاعات اضافی به مدل جلوگیری می شود. نمودار زیر به خوبی ن بشان میدهد که دقت الگوریتم ماشین بردار پشتیبان که با کمک الگوریتم بهینه سازی ازدحام ذرات بالاتر از سایر الگوریتم های می باشد و عملکرد بهتری را دارد:



در جدول زیر دقت الگوریتم های مختلف در هر دو بخش تست و ترین نشان داده شده که الگوریتم پیشنهادی ما دقت در هر دو بخش نسبت به سایر الگوریتم ها با دقت نزدیک به ۹۹ دو هر دو بخش عملکرد بهتری نسبت به سایر الگوریتم ها داشته:

	dt_gini	dt_entropy	knn	rf	lr	mlp	poly_svm	linear_svm	sigmoid_svm	rbf_svm	pos_rbf_svm
train_scores	0.921303	0.921303	0.967209	0.989258	0.928991	0.982135	0.960199	0.928087	0.832542	0.954545	0.999869
test_scores	0.928087	0.928087	0.964722	0.991407	0.925825	0.977838	0.959294	0.927635	0.843962	0.953415	0.995499

پیشنهادات کاربردی:

با استفاده از نتایج حاصل از این پژوهش می توان الگوریتم های مناسب و همچنین ویژگی های که بیشترین تاثیر بر سالم بودن با نبودن یک وب سایت را دارند شناسایی کرده و از آنها برای توسعه سامانه جهت افزایش امنیت کاربران در مقابل حملات فیشینگ استفاده کرد.

پیشنهاد آتی:

با استفاده از هوش مصنوعی در فضای وب می توان امنیت کاربران را به طور قابل توجه ای افزایش داد که موضوع سبب اعتماد بیشتر کاربران به کسب کار ها و ... می شود از آنجا که تعداد قربانیان حملات فیشینگ بسیار زیاد می باشد توسعه سسیتم های برای شناسایی حملات فیشینگ بسیار ضروری می باشد، پیشنهاداتی برای توسعه سیستم های جهت شناسایی حملات فیشینگ به شرح زیر می باشند:

۱- سامانه متشکل از دو بخش توسعه داده شود بخش اول وظیفه استخراج و بهینه سازی ویژگی های وب سایت را دارد و ویژگی های که استخراج می کند را در اختیار بخش دوم قرار می دهد بخش دوم الگوریتم یادگیری جمعی بوستینگ[۴۸]است که مجموعه ای ازالگوریتم های مختلف طبقه بند شامل می شود که سعی می کنندتشخیص دهند که ایا صفحه مورد نظر یک صفحه سالم است یا خیر و در صورت ناسالم بودن صفحه از طریق مرورگر به کاربر اخطار داده شود.

۲- مدلی جهت تشخیص صفحات فیشینگ با استفاده از ترکیب شبکه های عصبی با الگوریتم ژنتیک توسعه داده شود.

محدودیت ها:

۱- نداشتن سخت افزار مناسب برای اجرای سریع برنامه ها

۲- دسترسی محدود به سرویس های مانند گوگل کولب

۳- زمان بر بودن انجام محاسبات

REFERENCES:

[۴۸] Boosting

١-Korkmaz, M., Kocyigit, E., Sahingoz, O. K., & Diri, B. (2020). Deep Neural Network Based Phishing Classification on a High-Risk URL Dataset. In *SoCPaR* (pp. 648-657).

٢-Salem, O., Hossain, A., & Kamala, M. (2010, June). Awareness program and ai based tool to reduce risk of phishing attacks. In *2010 10th IEEE International Conference on Computer and Information Technology* (pp. 1418-1423). IEEE.

٣-Alsariera, Y. A., Adeyemo, V. E., Balogun, A. O., & Alazzawi, A. K. (2020). Ai meta-learners and extra-trees algorithm for the detection of phishing websites. *IEEE Access*, *8*, 142532-142542.

٤-Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. *arXiv preprint arXiv:1802.03162*.

٥-Miyamoto, D., Hazeyama, H., & Kadobayashi, Y. (2008, November). An evaluation of machine learning-based methods for detection of phishing sites. In *International Conference on Neural Information Processing* (pp. 539-546). Springer, Berlin, Heidelberg.

6-Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: a literature survey. *IEEE Communications Surveys & Tutorials*, *15*(4), 2091-2121.

7-Cao, J., Li, Q., Ji, Y., He, Y., & Guo, D. (2016). Detection of forwarding-based malicious URLs in online social networks. *International Journal of Parallel Programming*, *44*(1), 163-180.

8-Vazhayil, A., Harikrishnan, N. B., Vinayakumar, R., Soman, K. P., & Verma, A. D. R. (2018). PED-ML: Phishing email detection using classical machine learning techniques. In *Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA)* (pp. 1-8). Tempe, AZ, USA.

9-Subasi, A., Molah, E., Almkallawi, F., & Chaudhery, T. J. (2017, November). Intelligent phishing website detection using random forest classifier. In *2017 International conference on electrical and computing technologies and applications (ICECTA)* (pp. 1-5). IEEE.

10-Mondal, S., Maheshwari, D., Pai, N., & Biwalkar, A. (2019, December). A Review on Detecting Phishing URLs using Clustering Algorithms. In *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)* (pp. 1-6). IEEE.

١١- امیری, & نادری. (٢٠٢٠). استفاده از شبکههای خودرمزنگار موازی به منظور تشخیص صفحات جعلی اینترنتی. *مجله نوآوری های فناوری اطلاعات و ارتباطات کاربردی*, (٢), ٤١-٥٠.

١٢- بهارلو, & یاری. بهبود روش شناسایی وب سایت فیشینگ با استفاده از دادهکاوی روی صفحات وب. *دوفصلنامه فناوری اطلاعات و ارتباطات ایران*, ٤٣ (٤٣), ٢٧.

١٣- صف آرا, & صباح نو. (٢٠٢٠). افزایش دقت شناسایی صفحات جعلی وب با استفاده از الگوریتم بهینه سازی کفتار و شبکه عصبی مصنوعی. *فصلنامه علمی-پژوهشی فرماندهی و کنترل*, ٣ (٤), ٧٢-٩١.

١٤- دادخواه مهدي, داورپناه جزي محمد, & سعيدي مباركه مجيد. ارائه رويكردي به منظور شناسايي و پيش بيني وب سايتهاي فيشينگ به وسيله الگوريتمهاي كلاس بندي بر اساس مشخصه هاي صفحات وب.

١٥- کمالی زاده, سلمان, & شاهمحمدی. ارزیابی روشهای شناسایی وبسایت فیشینگ. *فصلنامه علمی پژوهشهای اطلاعاتی و جنایی*, ١١ (٤١), ٩-٣٨.