بسم الله الرحمن الرحيم

دانشگاه دولتی بجنورد
دانشکده فنی و مهندسی ۲

پایان نامه کارشناسی رشته مهندسی کامپیوتر

شناسایی داده های غیر نرمال جهت پیدا کردن crawler و intrusion ها از روی log سرور با استفاده از الگوریتم های یادگیری ماشین

نگارش
حسین بیدکی ، پریسا مجرائی مقدم

استاد راهنما
جناب مهندس احمد اعتضادی

تیر ۱۴۰۱

چکیده

دیتای این پروژه مجموعهای از لاگهای سروری واقعی است که جمع آورد شده و برای انجام پروژه تحویل داده شده است که هدف پیاده سازی یک مدل غیرنظارتی برای تشـــخیص خزنده یا ناهنجاری بودن یا نبودن درخواســتهایی اســت که به این وبســایت زده میشـــــود. مــا در این پروژه روش هــای متعـددی اعم از Isolation Forest ، Local Outlier Factor ، Autoencoder ها و ... را پیاده سـازی و نتایج آنها را با هم مقایســـــه نمودیم. اتوانکودر ها بهترین نتایج را به همراه داشـــــــــتند از آن برای فازproduction اســتفاده نمودیم. این پروژه در دو فاز انجام شــد که فاز اول اغلب اختصـاص به تحلیل داده و EDA به منظور اســـتخراج و مهندســـی ویژگی داده شـــد. در فاز دوم نیز مدلهای مختلف غیر نظارتی پیاده سـازی و مقایسـه شـدند. مهمترین نکات در این گزارش نوشته شده اند. تمامی کد ها در فایل قرار داده شده و در گیت هاب نیز پوش شده است. علاوه برآن، کدها و مسـتندات مربوط به پروژه نهایی در این ریپازیتوری وجود دارند. برای درک بهتر روند پروژه و کارهایی که برای آن انجام شـــده پیشنهاد میشود تا اسلایدها را مطالعه کنید. نمودارها و توجیحاتی که برای کارهایی که انجام شـده در آن اسـت که در این گزارش موجود نیست.

کلیدواژه‌ها: تشخیص خزنده ، ماشین لرنینگ

فهرست نوشتار

مقدمه..... ۱

فعالیت های انجام شده..... ۲

تحلیل اکتشافی داده (EDA) ۲

بیشترین IP های مشاهده شده ۳

بیشترین مرورگرهای دیده شده..... ۴

مهندسی ویژگی..... ۴

مدل های غیرنظارتی..... ۵

پیاده سازی مدل Isolation Forest ۶

پیاده سازی مدل Local Outlier Factor ۷

پیاده سازی الگوریتم Auto-encoder ۸

ارزیابی مدل ۹

انتظارات ما از مدل ۹

گزارش دسته بندی ۱۰

نتیجه ۱۱

فهرست منابع ۱۲

فهرست جدول‌ها

جدول ۱ : ویژگیهای دیتاست استفاده شده..... ۲
جدول ۲ : ویژگیهای مهندسی شده و استخراج شده از دیتاست برای هر تشست..... ۴
جدول ۳ : مقادیر خطای auto-encoder با معماری‌های گوناگون..... ۹
جدول ۴ : میانگین مقادیر ویژگی های به برای هر ویژگی برای هر نشست و مقایسه آنها..... ۱۰
جدول ۵ : گزارش دستهبندی برای مطمئنترین..... ۱۰

فهرست تصویرها

تصویر ۱ : داده های پرت زودتر به برگ درخت میرسند..... ۶
تصویر ۲ : قسمتهای اصلی auto-encoder ها..... ۸

فهرست نمودارها

نمودار ۱ : تعداد درخواست برای هر IP ۳
نمودار ۲ : تعداد درخواست برای مرورگر هر user1agent ۴
نمودار ۳ : تفکیک دادههای پرت و غیرپرت با آستانه گذاری بصورت تجربی..... ۷
نمودار ۴ : مصورسازی دادگان پرت و نرمال به کمک PCA..... ۷
نمودار ۵ : نحوه عملکرد الگوریتم ۷

مقدمه

خزنده وب، یک برنامه رایانه‌ای است که توانایی مرور و ثبت اطلاعات را از وب‌سایت‌ها به صورت خودکار دارد. «خزنده وب» به چندین شکل مختلف تعریف می‌شود که برخی از آنان عنکبوت‌های وب ، فهرست سازان خودکار ، ربات‌های نرم‌افزاری خودکار و نرم‌افزارهای FOAF هستند به عنوان مثال موتورهای جستجوگر با بهره‌گیری از این گونه نرم‌افزارها به صورت خودکار صفحات مختلف وب سایت‌ها را ثبت، آنالیز و رده‌بندی می‌کنند.

هنگامی که میخواهیم بر روی log سرور داده ها را آنالیز کنیم با وجود یوزر های غیر واقعی که همان خزنده ها هستند، با خطر فاقد اعتبار بودن نتایج روبرو هستیم.

از این رو هدف پروژه راه اندازی یک سیستم آفلاین برای تشخیص ناهنجاری و خزنده بر بستر وب به کمک الگوریتمهای غیرنظارتی است. دیتایی در فاز آموزش مدل در اختیار داشتیم، لاگ های جمع آوردی شده یک سروروواقعی است. نکته مهمی که این پروژه را چالشی میکرد، عدم وجود برچسب برای هر رکورد از دیتاست بود. در این فاز، پیش پردازش های لازم روی دیتای خام و فرآیند EDA بصورت متمرکز انجام شد. قبل از بررسی، پیشنهاد میشود فایل ارائه مطالعه شود. همچنین تمامی کدها و منابع پروژه در گیتهاب قابل مشاهده است.

فعالیت های انجام شده

کلیه فعالیت ها در طول انجام پروژه به صورت مختصر به شرح زیر است :

- تحلیل اکتشافی داده
- تمیزسازی داده
- تعریف و آموزش دادن مدل های غیرنظارتی پایه روی ویژگی های استخراج یافته
- مقایسه مدل های مختلف و ارزیابی
- آماده سازی اسلاید برای ارائه نهایی

تحلیل اکتشافی داده (EDA)

دیتاستی که در این پروژه استفاده شده متشکل از لاگ های جمع آوری شده یک سرور واقعی است که در فرمت یک فایل csv گردآوری گردیده است. تعداد رکوردهای این دیتاست، ۱۲۴۱۹۵۴۵ است.

ویژگیهای هر رکورد از این دیتاست عبارتند از:

ویژگی	جنس	توضیحات
ip	string	**به عنوان نمونه: 207.213.193.143**
time	string	**به عنوان نمونه: 2021-5-12T5:6:0.0+0430**
method	string	**Get/Put/Post/Options/Head**
status_code	integer	**وضعیت یک درخواست را نشان میدهد**
url	string	**به عنوان نمونه:images/fav_icon2.ico**
response_length	integer	**حجم اطلاعات موجود در آن درخواست.**
user_agent	string	**به عنوان نمونه : Googlebot-Image/۱.۰**
response_time	float	**زمان پاسخ سرور به کلاینت**

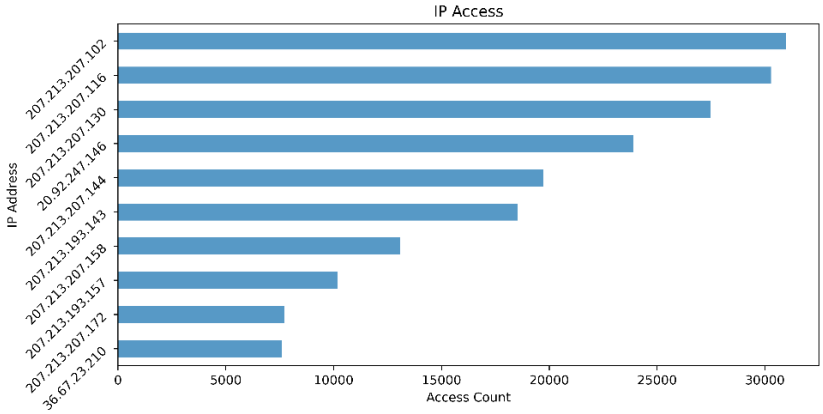
جدول ۱ ویژگیهای دیتاست استفاده شده

تحلیل اکتشافی داده یا همان EDA برای هر پروژهای که با داده سروکار دارد الزامی است.

EDA راهی برای مصورسازی، خلاصه‌سازی و تفسیرسازی اطلاعات مخفی و آشکار یک مجموعه دادگان است. EDA قدمی اساسی در علم داده است که با انجام آن یک دید خوبی از ویژگی های آماری یک دیتاست کسب میکنیم. با اتمام این مرحله میتوان از ویژگی های بدست آمده در یک روش نظارتی یا غیرنظارتی بهره برد.

ابتدا چند نمودار و شکل را رسم میکنیم تا شهود خوبی از دیتا کسب کنیم. به مرور میتوان از روی همین نمودار ها برخی از ویژگی های درخواستهای جعلی را فهمید.

بیشترین IP های مشاهده شده :



نمودار ۱ تعداد درخواست برای هر IP

سپس رکوردهای دیتاست را براساس بیشترین درخواست از آی‌پی ها مرتب میکنیم:

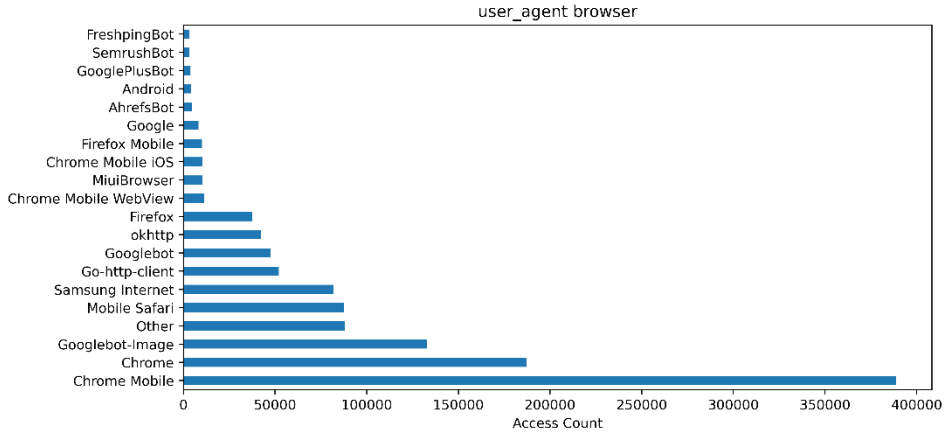
	ip	time	method	status_code	path	response_length	user_agent	response_time
0	207.213.207.102	2021-05-12 06:25:56+04:30	Get	304	cdn/articles/1148001967	0	Googlebot-Image/1.0	16.0
30968	207.213.207.116	2021-05-12 07:40:46+04:30	Get	304	cdn/profiles/1074674108	0	Googlebot-Image/1.0	16.0
61258	207.213.207.130	2021-05-12 09:25:34+04:30	Get	200	amp/price/1313296747	125767	Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Bu...	28.0
88748	20.92.247.146	2021-05-12 05:08:46+04:30	Get	200	js/profession.c67de06df71c34fc126d.js	107970	sentry/21.4.1 (https://sentry.io)	16.0
112660	207.213.207.144	2021-05-12 09:02:56+04:30	Get	200	amp/price/252451961	121639	Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Bu...	28.0
132378	207.213.193.143	2021-05-12 05:06:00+04:30	Get	304	cdn/profiles/1026106239	0	Googlebot-Image/1.0	32.0
150901	207.213.207.158	2021-05-12 09:25:55+04:30	Get	304	cdn/articles/2121333045	0	Googlebot-Image/1.0	16.0
163992	207.213.193.157	2021-05-12 05:07:04+04:30	Get	200	amp/blog/article/1197238235	101087	Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Bu...	144.0
174189	207.213.207.172	2021-05-12 09:28:50+04:30	Get	304	cdn/pro_photo_gallery/1540103160	0	Googlebot-Image/1.0	28.0
181890	36.67.23.210	2021-05-12 05:06:00+04:30	Head	200	877499224	0	Go-http-client/2.0	28.0

مشاهده میشود که اتفاقا IP هایی که بیشترین درخواستها از آنها آمده محتمل به جعلی بودن و یا برخاس ته از یک خزنده باش ند هس ت. زیرا با نگاه کردن به user_agent های آنها ش اهد نمونه هایی از بات ها مانند GoogleBot و ...هستیم.

بیشترین مرورگرهای دیده شده:

برای اینکه بفهمیم مرورگر کاربر چه بوده است از کتابخانه پایتونی به نام user_agents استفاده کردیم. با استفاده از این کتابخانه می توان ، اینکه درخواست از یک بات بوده یا نه، اینکه درخواست از یک کامپیوتر یا موبایل بوده و ... از روی user agent درخواست بدستآوریم.

در نمودار۲ میبینیم که برخی از مرورگرهایی که یک بات هستند ترافیک زیادی برای سایت ایجاد کرده اند.



نمودار ۲ تعداد درخواست برای مرورگر هر user1agent

مهندسی ویژگی

تا این قسمت شهود خوبی از ویژگیها و عوامل تاثیر گذار بر جعلی بودن یک درخواست از سمت کلاینت گرفته ایم. با توجه به دشوار بودن تشخیص anomaly بودن یا نبودن از روی تک درخواست، تصمیم بر این گرفتیم تا دیتاست را براساس IP و User agent گروهبندی کنیم. با این کار گویی داریم یک نشست تعریف میکنیم. پس در این مرحله سعی بر تعریف و مهندسیسازی ویژگی برای هر نشست داریم. تعریف ویژگی ها از یک مقاله پژوهشی الهام گرفته شده است و در جدول ۳-۳ لیست کامل آنرا میتوانید مشاهده کنید.

ویژگی برای نشست	توجیه و دلیل استفاده از این ویژگی
تعداد درخواست ها	زیادبودن تعداد درخواست در یک نشست موجب افزایش احتمال خزنده یا بات بودن آن کاربر میشود.
انحراف معیار عمق درخواستها	کاربرهای انسان، درخواست هایی که معمولا در یک نشست میزنند دارای عمق path با طول های متفاوتتر نسبت به خزندهها و باتها است.

درصد درخواستهای از خانواده ۴۰۰	معمولا خزندهها به پاسخهایی از سمت سرور با خطایی از خانواده ۴۰۰ بیشتر مواجه میشوند.
درصد درخواستهای از خانواده ۳۰۰	خانواده ۳۰۰ به معنای redirect شدن به یک صفحه دیگر است. خزندهها و باتها معمولا بیشتر با این پاسخ روبرو میشوند.
درصد درخواستهای با متد HEAD	در خزندهها یا بات ها بیشتر است زیرا احتمال مواجهه با صفحات پاکشده یا تاریخ گذشته در این نوع از کاربران زیادتر است.
درصد درخواستهای عکس	خزندهها و باتها معمولا به عکس ها درخواست نمیدهند.
جمع و میانگین response_lengthو response_time	بدلیل اینکه کاربرهای انسان از مرورگر برای دسترسی به صفحات وب استفاده میکنند، وقتی به یک صفحه درخواست میزنند، نشست مجبور به دریافت منابع و عکسهای متفاوتی است و همین باعث زیاد شدن زمان پاسخ و حجم درخواست میشود.
ویژگیهای استخراج شده از user_agent	مرورگر-سیستمعامل-بات بودن یا نبودن-از یک pc بودن یا نبودن.
تعداد درخواستهای به robtots.txt	خزنده ها برای اینکه متوجه شوند سرور سایت امکان خزش به چه صفحاتی را داده است، به robots.txtدرخواست میزنند. اینگونه اطلاعات در آنجا نوشته شده است.

جدول ۲ ویژگیهای مهندسی شده و استخراج شده از دیتاست برای هر تشست

مدل های غیرنظارتی

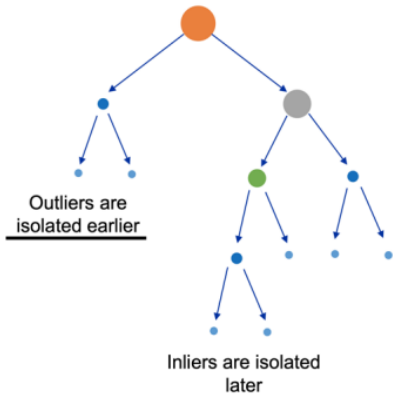
در این هفته از ویژگیهای تعریف شـــده در قســـمت قبلی اســـتفاده میکنیم تا مدلهای پایه را آموزش دهیم. به دلیل نداشـــتن برچســـب برای دادگان، باید از مدل های غیرنظارتی بهره میبردیم. مدل های غیرنظارتی زیادی در علم یادگیری ماشـــین برای تشـــخیص داده پرت تا به الان شـــناخته شـــده اند. از معروفترین آنها میتوان به موارد زیر اشاره نمود:

- Isolation Forest
- Local Outlier Factor
- One-class SVM
- Robust covariance

خوشـبختانه کتابخانه Scikit-learn این الگوریتم ها را پشـتیبانی و پیاده سـازی کرده اسـت و در این پروژه از آن اسـتفاده کردیم.کدها و نوتبوک های مربوط به این قسمت نیز در گیتهاب قابل مشاهده هستند.

پیاده سازی مدل Isolation Forest

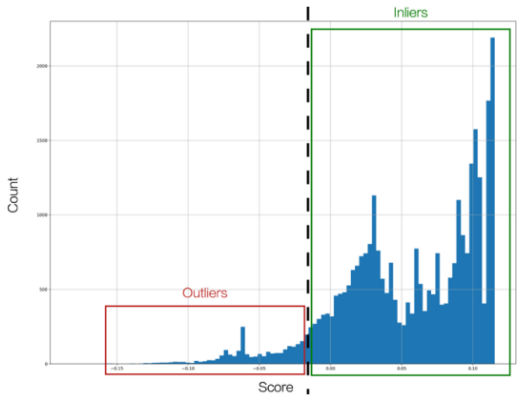
این روش، تکنیکی برای تشـــخیص دادههـای پرت بـه روش غیرنظـارتی اســـت. در این روش مشـاهدهها، با انتخاب ویژگیها به صـورت تصـادفی و جداسـازی مقدار أن به بیشـترین و کمترین مقادیر ویژگی انتخابی، ایزوله میشـوند. به دلیل خاصـیت بازگشـتی بودن این روش، این روش با یک سـاختار درختی قابل نمایش اســـــت. بدلیل اینکه مقادیر ویژگی های دادههای پرت به طرز قابل توجهی با بقیه دادگان تفاوت دارد، دادههای پرت زودتر در درخت تصـمیم ایزوله میشـوند(شـکل ۱-۳). به عبارتی با تنظیم کردن یـک آســـــتـانـه کـه خود یـک ابرپـارامتر برای تعـداد جـداســـــازی از بـالا تـا پـایین(برگ) درخـت میتوان دادههای پرت را شـــناســـایی کرد. پیادهســـازی ای که در Scikit-learn برای این روش شـده اسـت، یک امتیاز بین -۱ و ۱ با اسـتفاده از تعداد جداسـازی ها به هر نمونه از دیتا میدهد. هرچه امتیازبه -۱ نزدیک تر باشـد، به معنای پرت بودن(در اینجا خزنده یا بات بودن کاربر) میباشد.



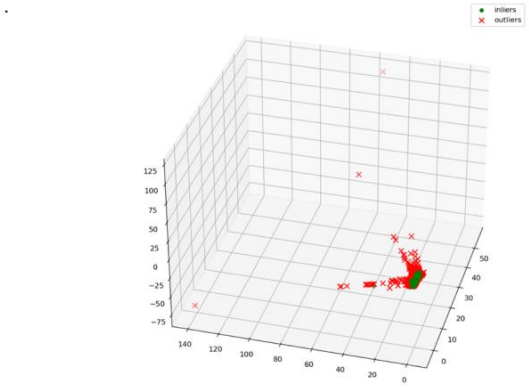
تصویر ۱ داده های پرت زودتر به برگ درخت میرسند

همانطور که گفته شـده ابرپارامتر آسـتانه باید توسـط ما تعیین شـود و با توجه به نداشـتن برچسـب برای دادگـان، امکـان fine-tune کردن این پـارامتر وجود نـدارد. پس در این مرحلـه فعلا بصـــورت تجربی و آزمونوخطـا این آستانه تعیین شـــد. در نمودار ۴ هیســتوگرام امتیازهای نمونه ها مشـاهده میشـود. در این حالت، نمونه های با امتیاز کمتر از ۰ به عنوان داده پرت در نظر گرفته شده اند.

با این آسـتانه گذاری از ۳۱۵۴۱ نشسـتی که با گروهبندی به کمک IP و User agent بدسـت آمده بود، ۶۱۵ تای آنها داده پرت تشـخیص داده شـدند. برای اینکه بفهمیم چقدر این مدل درسـت آموزش دیده، به کمک کتابخانه user_agent و ویژگی is_bot بودن آن دیدیم که ۶۰۳ تا از ۶۱۵ تایی که داده پرت تشـخیص داده شـدند، بات هسـتند. این به این معنا بود که نسـبتا مدل به عنوان یک مدل پایه عملکرد خوبی داشـــته اســـت. اما هنوز به صـــورت قطعی بدلیل عدم وجود برچســـب برای دادهگان نمیتوان دقت برای روش و مدل به کار گرفته شده مشخص نمود.



نمودار ۳ تفکیک دادههای پرت و غیرپرت با آستانه گذاری بصورت تجربی

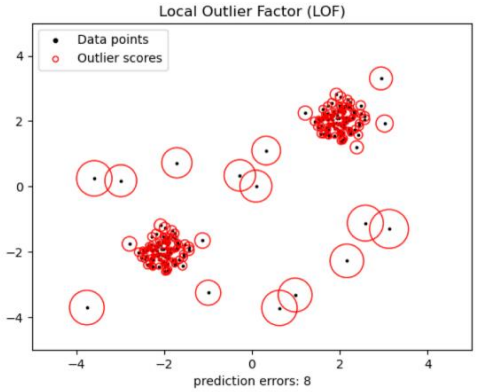


نمودار ۴ مصورسازی دادگان پرت و نرمال به کمک PCA

پیاده سازی مدل Local Outlier Factor

در روش LOF، انحراف محلی چگالی هر نمونه نسبت به همسایگان خود محاسبه میشود. پرت بودن یک داده به میزان ایزوله و تنها بودن آن نمونه نسبت به همسایگان خود سنجیده میشود.

همچنین چگالی محلی به کمک متریک فاصلهای که در روش KNN محاسبه میشود صورت میگیرد. در نمودار ۵-۳ نیز مشاهده میکنید که هرچه یک نمونه همسایگان کمتری داشته باشد، احتمال داده پرت بودن آن نیز بیشتر است.

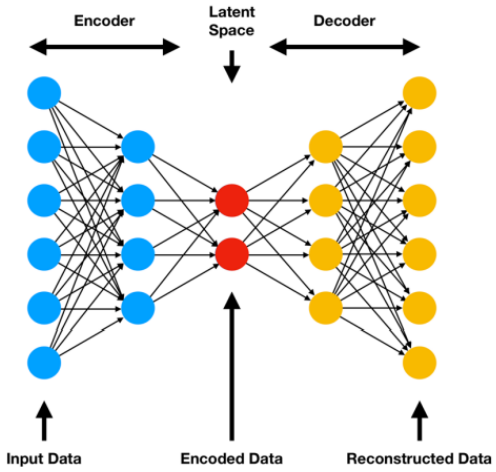


نمودار ۵ نحوه عملکرد الگوریتم

اما این روش نتایج خوبی برای دیتاستی که ما داشتیم به همراه نداشت زیرا حتی محتمل ترین نمونه هایی که به عنوان داده پرت درنظر گرفته بود گاهاً داده نرمال بودند. به همین دلیل از این روش در ادامه استفاده ای نشد.

پیاده سازی الگوریتم Auto-encoder

اتوانکودرها یک نوع خاص ی از ش بکههای عص بی هس تند که مقادیر نورونهای ورودی را در خروجی کپی میکنند. در فرآیند آموزش این نوع شبکه، دیگر نیازی به برچسب دادهها وجود ندارد برای همین میتوان اتوانکودر ها را جزو الگوریتمهای غیر نظارتی دانست. به طور معمول لایه های مخفی در این نوع شبکه ها، دارای تعداد نورون کمتری نسبت به نورونهای ورودی و خروجی دارند. به همین دلیل، لایه های مخفی اطلاعات ضروری را در خود ذخیره میکنند و از نویز ها صرف نظر میکنند. در فرآیند آموزش این نوع شبکه دو قسمت مهم encode و decode وجود دارد. در مرحلهای encoding، مقادیر ورودی را فشرده سازی میکند و به فضای برداری لایه مخفی میبرد. در مرحله decoding، اطلاعات فشردهشده، باز سازی میشوند. در شکل ۲-۳ نیز این مراحل مصور شده است.



تصویر ۲ قسمتهای اصلی auto-encoder ها

اتوانکودرها کاربردهای وسیعی در زمینه پردازش تصویر و بینایی ماشین دارند همچنین نتایج درخشانی در زمینه تشخیص ناهنجاری نیز داشته اند. در فرآیند decoding، هنگامی که عمل بازسازی صورت میگیرد میتوان خطای یکسان نبودن داده خروجی با داده ورودی اولیه را حساب کرد. به عبارت دیگر، لایه مخفی سعی بر یادگیری یک embedding از داده های ورودی است و میخواهد داده های ورودی را تنها با داشتن ویژگی های موجود در لایه مخفی بازسازی کند. به طبع اگر داده پرتی که اختلاف زیادی از نظر مقادیر ویژگیها با دادههای دیگر وجود داشته باشد،

خطای‌بازسازی بسیار زیاد و متفاوت است. پس با این رویکرد میتوان از اتوانکودر ها برای تشخیص ناهنجاری نیز استفاده نمود. در فرآیند آموزش از بهینه ساز Adam و تابع خطای MSE برای محاسبه خطا استفاده کردیم. همچنین از ReLu به عنوان تابع غیرخطی ساز استفاده کردهایم. دیتا ست با توزیع ۸۰-۲۰ به دو داده آموزش و تست تقسیم شد و در جدول مقادیر خطا پس از آموزش برای شبکه های با معماری هایی متفاوت قابل مشاهده است.

تعداد نورونها	خطای داده آموزش	خطای داده تست
[15, 7, 15]	٤٢.٠	٤٨.٠
[15, 3, 15]	**٢٨.٠**	**٣٩.٠**
[15, 7, 3, 7, 15]	٢٩.٠	٤٣.٠
[15, 7, 7, 7, 15]	٣١.٠	٤٢.٠

جدول ۳ مقادیر خطای auto-encoder با معماری‌های گوناگون

پس از آموزش مدل، برای فاز پیشبینی ناهنجاری‌بودن یا نبودن، مشابه الگوریتمهای قبلی، باید یک حد آستانه برای خطای MSE نیز تعریف کرد. با توجه به دانش قبلی و بررسی بات بودن یا نبودن نشست های داخل دیتافریم براساس user agent شان، حدود ۵ درصد از دیتاست به واضح بات بودند، بنابراین آستانه خطای MSE به طوری انتخاب شد که ۵ درصد از نمونه ها به عنوان ناهنجاری تشخیص داده شوند.

ارزیابی مدل نهایی

اتوانکودر با توجه به اینکه عملکرد بهتری روی دیتاست داشت به عنوان مدل نهایی انتخاب شد و تصمیم نهایی بر این شد از این مدل برای خوشه بندی استفاده شود .

انتظارات ما از مدل

در مرحله EDA چندین ویژگی آماری تعریف و مهندسی شدند. به طور مثال یکی از آنها تعداد درخواستهای نشست بود و ما توقع داشتیم که برای نشستهای جعلی، تعداد درخواست ها به طور قابل توجهی بیشتر باشد. پس از پیشبینی مدل اتوانکودر، میانگین تعداد درخواستها در نشستهای نرمال و همچنین در نشست های جعلی را مقایسه نمودیم و دقیقا همان چیزی که انتظار

میرفت رخ داد. در جدول زیر این مقایس ها قابل مشـــاهده هســـتند که فقط دو تا از ویژگیهای تعریف شده نتیجه مطلوبی از خود نشان ندادند:

Average of	# of requests	Path length STD	Percentage of 4xx	Percentage of 3xx	Percentage of HEAD reques	consecutive repeated requests	robots.txt requests	Percentage of image requests
Outliers	231	0.43	3.21%	9.33%	0.34%	0.81	0.08	9.74%
Inliers	25	0.39	0.68%	26%	0.00003	0.62	0.0	28.16%

جدول ۴ میانگین مقادیر ویژگی های به برای هر ویژگی برای هر نشست و مقایسه آنها

گزارش دسته بندی

برای اینکه ارزیابی دیگری داشـته باشـیم، ۱۵۰ تا از مطمئنترین پیشبینی های مدل را به صورت دستی برچسب زدیم و ارزیابی های کلاسیک مدلهای نظارتی را برای آنها اعمال نمودیم:

Accuracy	90%
Precision	85.71%
Recall	100%
F1-score	92.30%

جدول ۵ گزارش دستهبندی برای مطمئنترین

نتیجه

بهترین و معروفترین الگوریتمهای غیر نظارتی که در سالهای اخیر نتایج بسزایی در تمامی زمینه ها از خود نشان دادند مورد مطالعه و پژوهش قرار گرفتند. توانستیم یک سیستم که قابلیت تشخیص جعلی بودن یا نبودن درخواست هایی که توسط کاربران به یک وبسایت زده میشود را با دقت خوبی داشته باشد، پیاده سازی کنیم. در این پروژه از اتوانکودرها به عنوان یک شبکه عصبی با یک رویکرد غیرنظارتی در عین حال دقت بالا، استفاده نمودیم.

فهرست منابع

Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters, †NICTA Victoria Laboratory Department of Computer Science and Software Engineering The University of Melbourne Parkville, Victoria 3010 Australia

واژنامه انگلیسی به فارسی

crawler : خزنده

intrusion : نفوذ

anomaly : ناهنجاری

University of bojnurd

Identify abnormal data to find crawlers and intrusions from the log server using machine learning algorithms

By
Hosein bidaki , Parisa MajraeiMoghadam

Supervisor
Mr. Ahmed Etezadi

July 2022