دانشکده مهندسی کامپیوتر

بهبود سامانه‌های تشخیص نفوذ در شبکه‌های نسل ۵ با استفاده از تولید داده مصنوعی و شبکه‌های بر پایه مبدل

پروژه کارشناسی مهندسی کامپیوتر

سینا اسکندری

استاد راهنما

دکتر وصال حکمی

مهر ۱۴۰۲


بسم الله الرحمن الرحيم

چکیده

در دنیای امروزه، حملات در شبکه بسیار فراگیر شده‌اند و می‌توانند پیامدهای فاجعه‌باری برای سازمان‌ها و افراد داشته باشند. به‌خصوص در شبکه‌های نسل ۵ که در حال توسعه و فراگیری می‌باشند و امنیت این شبکه‌ها امری با اهمیت تلقی می‌شود. این فعالیت‌های مخرب، اغلب از آسیب‌پذیری‌ها در شبکه سوءاستفاده می‌کنند و باعث می‌شوند که توسعه اقدامات امنیتی مقاوم امری ضروری به حساب آید. سامانه‌های تشخیص نفوذ به عنوان یک سازوکار دفاعی در برابر این تهدیدها توسعه‌یافته‌اند و طراحی شده‌اند که دسترسی‌های غیرمجاز، رفتارهای غیرعادی و انواع مختلف حمله را نظارت و شناسایی کنند. با این حال، اثربخشی این سامانه‌ها به کیفیت و کمیت داده‌های آموزشی موجود بستگی دارد. بسیاری از مجموعه داده‌های تشخیص حمله موجود، از مشکل نامتوازنی[۱] کلاس‌ها رنج می‌برند که در آن‌ها انواع خاصی از حمله‌ها کمتر موجود هستند و باعث ایجاد مدل‌های جانبدارانه و کاهش عملکرد سامانه می‌شود.

هدف این پروژه این است که با استفاده از روش‌های یادگیری عمیق، علی‌الخصوص شبکه‌های مولد متخاصم[۲] برای تولید داده‌ی مصنوعی و مدل‌های مبدل[۳] برای دسته‌بندی، چالش نامتوازنی مجموعه داده را برطرف کند و به دقت بالاتری در مسئله دسته‌بندی دست یابد.

برای برطرف کردن مشکل نامنتوازنی مجموعه داده، مدلی تحت عنوان مدل مولد متخاصم طراحی می‌کنیم و روی مجموعه داده آموزش می‌دهیم. این مدل آموزش می‌بیند که الگوها و ویژگی‌های هر دو کلاس اکثریت و اقلیت را بدست آورد و با استفاده از آن‌ها مجموعه داده را متوازن کند.

داده مصنوعی تولید شده به مجموعه داده اصلی اضافه می‌شود و با بهره‌گیری از مدل‌های مدرن بر پایه معماری مبدل‌ها، که روی داده جدید آموزش می‌بینند، دقت مسئله دسته‌بندی را بهبود می‌دهیم و به سامانه تشخیص حمله مقاوم‌تری دست می‌یابیم.

واژگان کلیدی: سامانه تشخیص حمله در شبکه، شبکه نسل ۵، یادگیری عمیق، شبکه مولد متخاصم، مبدل، نامتوازنی مجموعه داده

---

[1] Unbalanced
[2] Generative adversarial networks
[3] Transformer Models

فهرست مطالب

فهرست تصاویر

فهرست جداول

فصل ۱

مقدمه

۱-۱ شرح مسئله

با وجود اینکه اینترنت امکانات و فرصت‌های زیادی را برای مردم ایجاد می‌کند، به جنایتکاران این اجازه را می‌دهد که حملات غیر قانونی در شبکه انجام دهند که باعث ضررهای زیادی شامل شود. تا سال ۲۰۱۷، این حملات باعث ضرری ۶۰۰ میلیارد دلاری شدند.[۲۲] محققان برای جلوگیری از این تهدیدها سامانه‌های تشخیص نفوذ شبکه (NIDS[1]) را پیشنهاد دادند. این سامانه‌ها از نفوذ به شبکه جلوگیری می‌کنند و یکپارچگی [۲] ، محرمانگی[۳] و دسترس‌پذیری[۴] آن را حفظ می‌نمایند. روش‌های یادگیری ماشین [۵] مانند ماشین بردار پشتیبان (SVM[6]) و جنگل تصادفی [۷] استفاده شده‌اند که فعالیت‌های مخرب در شبکه را تشخیص دهند. اما با توجه به اینکه در این مسئله حجم داده زیاد می‌باشد و ویژگی‌های آن‌ها پیچیده می‌باشد، این رویکردها نمی‌توانند حملات شبکه را به طور موثر تشخیص دهند.[۱۵]

---

[1] Network Intrusion Detection System
[2] Integrity
[3] Confidentiality
[4] Availability
[5] Machine Learning
[6] Support Vector Machine
[7] Random Forest

به علت پیشرفت یادگیری عمیق [1] در زمینه‌های پردازش زبان‌های طبیعی [2] و پردازش تصویر [3] در سال‌های اخیر ، توجه بسیاری در بحث امنیت رایانه‌ای [4] به این رویکردها شده است.[۹] شبکه نسل ۵ که یکی از جدیدترین فناوری‌ها در ارتباطات بی‌سیم است، احتیاج دارد که در مقابل حملات مقاوت نشان دهد و در صورت بروز حمله، توانایی شناسایی و جلوگیری از آن‌ها را داشته باشد.

۱-۲ ساختار گزارش

در فصل دوم گزارش به بررسی مجموعه داده مورد استفاده می‌پردازیم و ویژگی‌ها و خصوصیات آن را بیان می‌کنیم، انواع حملات در آن را بررسی می‌کنیم، نحوه آماده‌سازی داده برای آموزش مدل بیان می‌کنیم و به مشکل نامتوازنی در مجموعه داده می‌پردازیم. در فصل سوم به مشکل نامتوازنی در مجموعه داده‌ها پرداخته می‌شود و راه‌حل استفاده شده در این پروژه برای مقابله با آن ارائه می‌شود. در فصل چهارم، ابتدا پژوهش‌های پیشین در زمینه دسته‌بندی داده‌ها بررسی می‌شوند، سپس معیارهای ارزیابی در یادگیری ماشین و عمیق معرفی می‌شوند و در نهایت مدلی بروز و مدرن برای دسته‌بندی داده‌ها معرفی می‌شود.

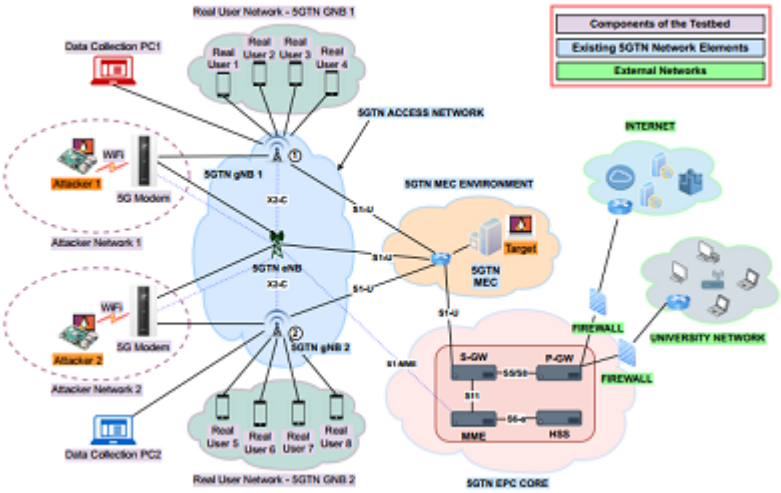
[1] Deep Learning
[2] Natural Language Processing
[3] Image Processing
[4] Cybersecurity

فصل ۲

آشنایی با مجموعه داده 5G-NIDD

برای آموزش یک مدل یادگیری عمیق احتیاج به یک مجموعه داده مناسب است که مدل الگوها و ویژگی‌های داده را استخراج کند. مجموعه داده‌های متنوعی برای آموزش NIDS ها مانند KDD99[۴]، NSL-KDD[۲۶] و CICIDS2017[۳۶] موجود هستند. در این پروژه از مجموعه داده 5G-NIDD [۳۳] استفاده می‌کنیم که اخیراً منتشر شده و اولین مجموعه داده ترافیک 5G برای تشخیص حمله می‌باشد. اغلب مجموعه داده‌های موجود از شبکه‌های مجازی شده [۱] که به خصوص برای ایجاد یک مجموعه داده شده، استفاده می‌کنند و این امر سبب می‌شود که تنظیمات و بستر آزمایشی[۲] دنیای واقعی[۳] برای ارزیابی سامانه‌ها امری حیاتی باشد.[۲۹] با توجه به این نکته، 5G-NIDD با استفاده از یک شبکه کاربردی نسل ۵ واقع در فنلاند جمع‌آوری شده و بسیار شبیه به سناریو یک شبکه واقعی دارد. [۳۳] 5G-NIDD از ترکیبی از ترافیک حمله و بی‌خطر که شامل پروتکل‌های مختلف از جمله HTTP، HTTPS، SSH و SFTP و سناریوهای مختلف حمله به وجود آمده که انواع این حملات را در ادامه بررسی می‌کنیم. بستر آزمایشی این مجموعه داده در تصویر ۲-۱ آمده است.

[1] Virtualized networkrs
[2] Testbed
[3] Real-World



شکل ۲-۱: بستر آزمایشی مجموعه داده 5G-NIDD[۳۳]

۲-۱ انواع حملات در مجموعه داده 5G-NIDD

مجموعه داده 5G-NIDD از ۸ نوع حمله متفاوت تشکیل شده که به شرح زیر می‌باشند.

۱. ICMP Flood: در این روش که از ابزار Hping3 استفاده می‌شود، درخواست‌های ICMP echo با شدت بالا به هدف ارسال می‌شود و باعث می‌شوند که کاربران عادی نتوانند از سرویس‌ها استفاده کنند.

۲. UDP Flood: در این نوع حمله، تعداد زیادی بسته UDP به هدف ارسال می‌شود و باعث قطع دسترسی کاربران می‌شود. چون پروتکل UDP غیراتصال‌گرا[۱] می‌باشد، امکان ارسال بسته‌ها با تعداد بالا وجود دارد.

۳. SYN Flood: در این روش از دست‌دادن سه طرفه[۲] پروتکل TCP سوءاستفاده می‌شود. در اتصال TCP، ابتدا مبدأ یک بسته SYN برای شروع ارتباط به مقصد ارسال می‌کند، سپس مقصد یک بسته SYN ACK در جواب SYN به مبدأ می‌فرستد و در نهایت مبدأ یک بسته ACK به مقصد می‌فرستد تا دست دادن سه طرفه را تکمیل کند. در حمله SYN Flood ، حمله‌کننده مرحله آخر را انجام نمی‌دهد و اتصال را نیمه‌باز رها می‌کند. تعداد زیاد این اتصالات نیمه‌باز، باعث از دسترس خارج شدن سرویس می‌شود.

[1] Connection-less
[2] Three Way Handshake

۴. HTTP Flood: این نوع حمله لایه کاربرد[1] را مورد هدف قرار می‌دهد. این روش، یک روش محبوب برای حمله محروم‌سازی از سرویس(DoS[2]) می‌باشد، چون توانایی شبیه‌سازی رفتار انسان را دارد و شناسایی آن مشکل می‌گردد.

۵. Slowrate DoS: یک روش سوءاستفاده دیگر از لایه کاربرد می‌باشد که از حمله‌های نرخ پایین ولی طولانی مدت استفاده می‌کند. به طور مثال، در سرآیند[3] بسته حجم بسته ارسالی را بیشتر از حجم واقعی اعلام می‌کنند و گیرنده منتظر دریافت بسته با حجم اعلام شده می‌ماند در حالی که چیزی ارسال نمی‌شود. به علت پایین بودن نرخ این حمله‌ها، شناسایی آن‌ها دشوار می‌گردد.

۶. Port Scan: این روش‌ها معمولاً پیش از انجام حمله واقعی انجام می‌شوند و هدف آن‌ها دریافت اطلاعات در مورد درگاه[4]های هدف و پیدا کردن فرصت برای حمله است. در این مجموعه داده ۳ روش حمله از این دسته وجود دارد که به شرح زیر می‌باشند.

- SYN Scan
- TCP Connect Scan
- UDP Scan

۲-۲ آماده‌سازی و پیش‌پردازش داده

برای اینکه مجموعه داده خود را برای آموزش مدل یادگیری عمیق آماده کنیم و توانایی مدل را برای برای یادگیری الگوهای معنادار افزایش دهیم، یک سری مراحل اولیه پیش‌پردازش[5] را برای داده خام اعمال می‌کنیم.

[1] Application Layer
[2] Denial of Service
[3] Header
[4] Port
[5] Preprocessing

۱. تبدیل بسته‌ها به جریان شبکه[1]

دو رویکرد اصلی سامانه‌های تشخیص نفوذ شبکه شامل شیوه مبتنی بر بسته[2] و مبتنی بر جریان[3] می‌باشد. در روش مبتنی بر بسته، اطلاعات سرآیند و محتویات هر بسته که در شبکه رد و بدل می‌شود مورد بررسی قرار می‌گیرد. این روش بار محاسباتی بسیار زیادی دارد و پیاده‌سازی آن‌ها در شبکه‌هایی که مقیاس بزرگ دارند، دشوار است. روش‌های مبتنی بر جریان خلاصه‌ای از اطلاعات را بر اساس دنباله‌ای از بسته‌ها که بین دو نقطه پایانی جابه‌جا می‌شوند، بررسی می‌کنند.[۲۴]

Netflow یک پروتکل نظارتی و جمع آوری اطلاعات ترافیک شبکه مبتنی بر جریان است که توسط Cisco توسعه یافته است.[۱۲] بسته‌های جمع‌آوری شده در 5G-NIDD به صورت فایل‌های pcap ذخیره شده‌اند و لازم است که به Netflow تبدیل شوند. برای این کار از ابزار Argus[۳۱] استفاده می‌کنیم. این جریان داده‌ها شامل یک سری ویژگی[4] مانند آدرس IP مبدأ و مقصد، نوع پروتکل ارتباط و مدت زمان جریان می‌باشد. لیست همه‌ی ویژگی‌های قابل استخراج در این آدرس موجود است.

۲. رسیدگی به مقادیر گم‌شده

یکی از مراحل مهم در پیش‌پردازش مجموعه داده‌ها رسیدگی به مقادیر گم‌شده می‌باشد، زیرا وجود این مقادیر روی عملکرد مدل تأثیر دارد.[۲۸] ویژگی‌ها در مجموعه داده به دو دسته ویژگی‌های پیوسته[5] و طبقه‌بندی‌شده[6] تقسیم می‌شوند. نحوه رسیدگی به مقادیر ویژگی‌های گم‌شده، به نوع ویژگی و تعداد داده‌هایی که آن ویژگی را ندارند، بستگی دارد. به طور مثال اگر درصد کمی از داده‌ها یک ویژگی را نداشته باشند، می‌توان آن‌ها را حذف کرد ولی اگر درصد زیادی از داده‌های آن ویژگی را نداشته باشند، می‌توان آن ویژگی را به طور کلی نادیده گرفت و حذف کرد. یک روش دیگر برای ویژگی‌های پیوسته این است که مقادیر گم‌شده را با میانگین، میانه و یا مد بقیه مقادیر جایگزین کنیم. برای ویژگی‌های طبقه‌بندی شده نیز می‌توان یک کلاس جدید تحت عنوان خالی به کلاس‌ها اضافه کنیم. در جدول ۲-۱ نحوه رسیدگی به ویژگی‌های گم‌شده مشخص شده‌اند.

[1] Netflow
[2] Packet-Based
[3] Flow-Based
[4] Feature
[5] Continuous
[6] Categorical

نام ویژگی	نوع ویژگی	درصد گم‌شدگی	نحوه رسیدگی
sTos	پیوسته	0.01	جایگزینی با میانگین
dTos	پیوسته	78	حذف ویژگی
sDSb	طبقه‌بندی شده	0.01	اضافه کردن کلاس
dDSb	طبقه‌بندی شده	78	حذف ویژگی
sTtl	پیوسته	0.01	جایگزینی با میانگین
dTtl	پیوسته	78	حذف ویژگی
sHops	پیوسته	0.01	جایگزینی با میانگین
sTtl	پیوسته	0.01	جایگزینی با میانگین
SrcGap	پیوسته	77	حذف ویژگی
DstGap	پیوسته	77	حذف ویژگی
SrcWin	پیوسته	80	حذف ویژگی
DstWin	پیوسته	85	حذف ویژگی
sVid	پیوسته	91	حذف ویژگی
dVid	پیوسته	99	حذف ویژگی
SrcTCPBase	پیوسته	77	حذف ویژگی
SrcTCPBase	پیوسته	81	حذف ویژگی

جدول ۲-۱: نحوه رسیدگی به ویژگی‌هایی که مقادیر گم‌شده دارند.

۳. رمزگذاری[1] ویژگی‌های طبقه‌بندی شده

برای اینکه بتوان ویژگی‌های طبقه‌بندی شده را به عنوان ورودی به یک شبکه مصنوعی[2] داد، احتیاج هست که مقادیر آن‌ها را به مقادیر عددی رمزگذاری کنیم. دو روش اصلی برای این کار رمزگذاری ترتیبی[3] و یک‌داغ[4] می‌باشد[۳۰].

در روش ترتیبی، هر کلاس به یک عدد نگاشت می‌شود. مشکل این روش این است که کلاسی که به طور مثال به عدد ۳ نگاشت شده است، ۳ برابر کلاسی که به عدد ۱ نگاشت شده، ارزش دارد.

در روش یک‌داغ، اگر تعداد حالات یک ویژگی برابر d باشد، به ازای هر حالت d متغیر تعریف می‌شود که 1 - d از آن‌ها برابر صفر و یکی از آن‌ها برابر یک است. به طور مثال برای یک ویژگی سه حالته، حالت اول به 001، حالت دوم به 010 و حالت سوم به 100 نگاشت می‌شوند. مزیت این روش این است که حالت‌های مختلف با هم فاصله یکسان دارند و ما نیز از این روش استفاده می‌کنیم.

[1] Encoding
[2] Neural Network
[3] Ordinal
[4] One-hot

۴. تغییر مقیاس ویژگی‌ها[۱]

تغییر مقیاس ویژگی روشی است که برای بهنجار کردن[۲] محدوده متغیرهای مستقل یا ویژگی های داده‌ها استفاده می شود. از آنجایی که دامنه مقادیر در ویژگی‌های مختلف متفاوت است، ممکن است توابع هدف[۳] بدون بهنجارسازی به درستی عمل نکنند. یک دلیل دیگر برای تغییر مقیاس ویژگی‌ها این است که سرعت همگرا شدن الگوریتم گرادیان کاهشی[۴] با تغییر مقیاس ویژگی بسیار سریع‌تر از بدون آن همگرا می شود.[۳۸] برای ویژگی‌های مجموعه داده، از روش تغییر مقیاس استاندارد استفاده می‌کنیم که مقیاس داده‌ها را به گونه‌ای تغییر می‌دهد که میانگین و واریانس آن‌ها به ترتیب برابر ۰ و ۱ شود.

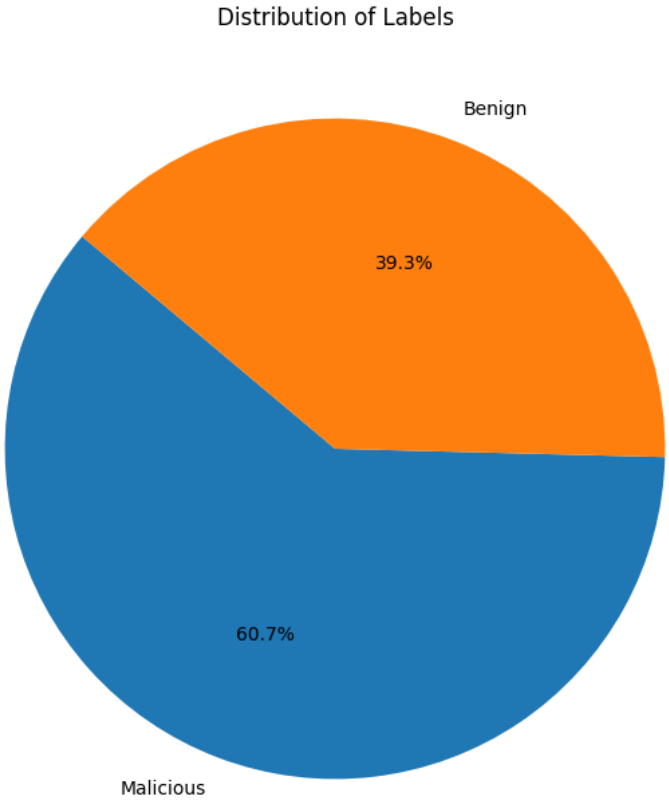
۵. جدا کردن مجوعه داده آموزش[۵] و آزمایشی[۶]

در مسائل یادگیری ماشین عرف است که مجموعه داده را به دو بخش آموزش و آزمایشی تقسیم کنیم و در مرحله آموزش فقط از مجموعه داده آموزشی استفاده کنیم و با استفاده از مجموعه داده آزمایشی عملکرد مدل را ارزیابی کنیم. علت این کار این است که مقاومت مدل در مقابل داده‌ای که تا به حال ندیده است(شرایط دنیای واقعی) سنجیده شود. در گذشته که حجم مجموعه داده‌ها کم(۱۰۰ یا ۱۰۰۰ یا ۱۰۰۰۰) بود، معمولاً نسبت تقسیم داده به صورت ۸۰ به ۲۰ یا ۷۰ به ۳۰ بود. در حال حاضر حجم مجموعه داده‌ها افزایش یافته و به اعداد میلیونی رسیده است و اختصاص دادن حجم زیادی از داده برای آزمایش، کاری بیهوده است، زیرا مجموعه داده آزمایش باید فقط به اندازه‌ای بزرگ باشد که بتوان الگوریتم‌های مختلف را روی آن‌ها ارزیابی و بهترین را انتخاب کرد و همچنین داده آموزشی بیشتر و متنوع‌تر می‌تواند عملکرد مدل را بهبود دهد.[۲۷] در مجموعه داده 5G-NIDD حدود ۱.۲ میلیون نمونه داده موجود است و در این پروژه آن را به نسبت ۹۷ به ۳ تقسیم می‌کنیم.

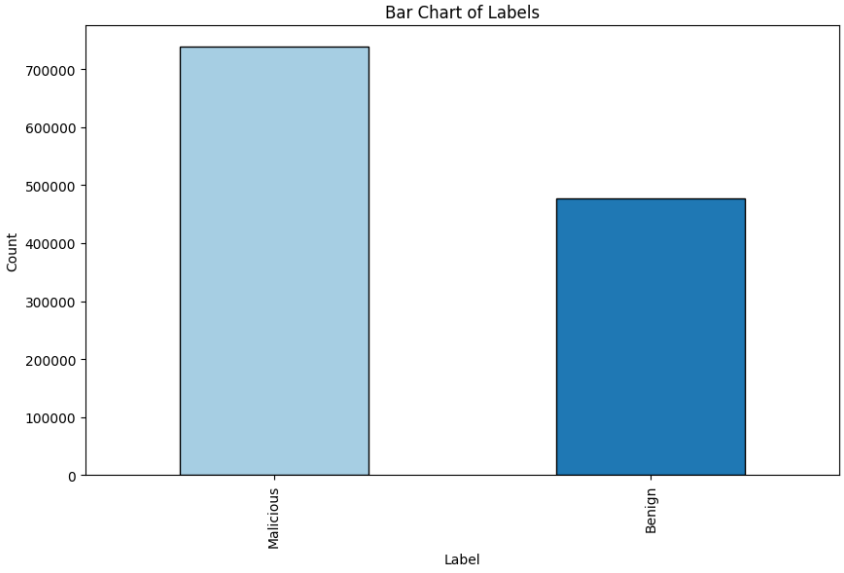
[1] Feature Scaling
[2] Normalize
[3] Objective Function
[4] Gradient Descent
[5] Train
[6] Test

۲-۳ اطلاعات آماری مجموعه داده 5G-NIDD

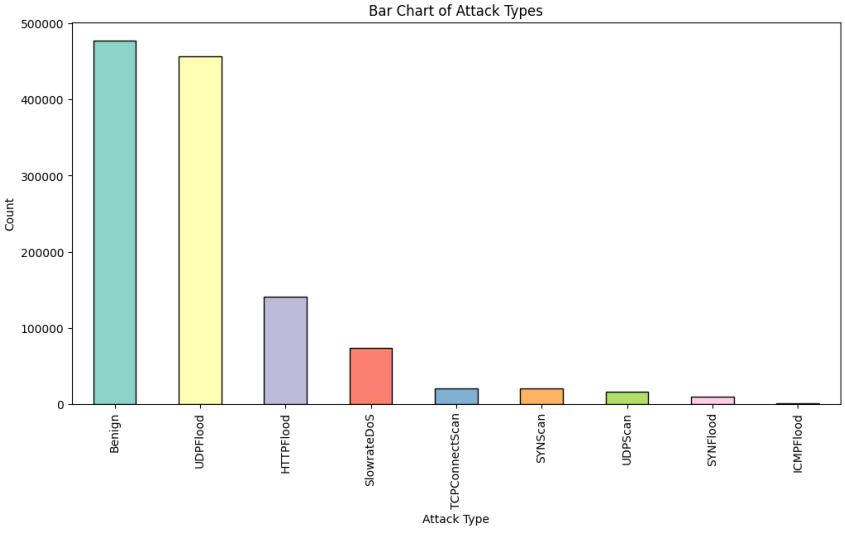
در این بخش به بررسی مجموعه داده می‌پردازیم و نمودارهایی رسم می‌کنیم که دید وسیع‌تری نسبت به مجموعه داده داشته باشیم. این نمودارها در شکل‌های ۲-۲ تا ۲-۱۱ قابل مشاهده هستند.



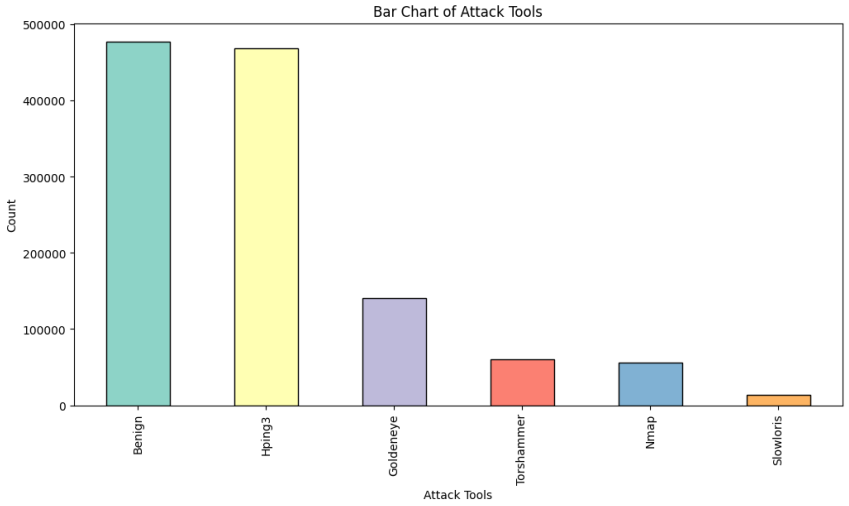
شکل ۲-۲: نسبت برچسب داده‌ها



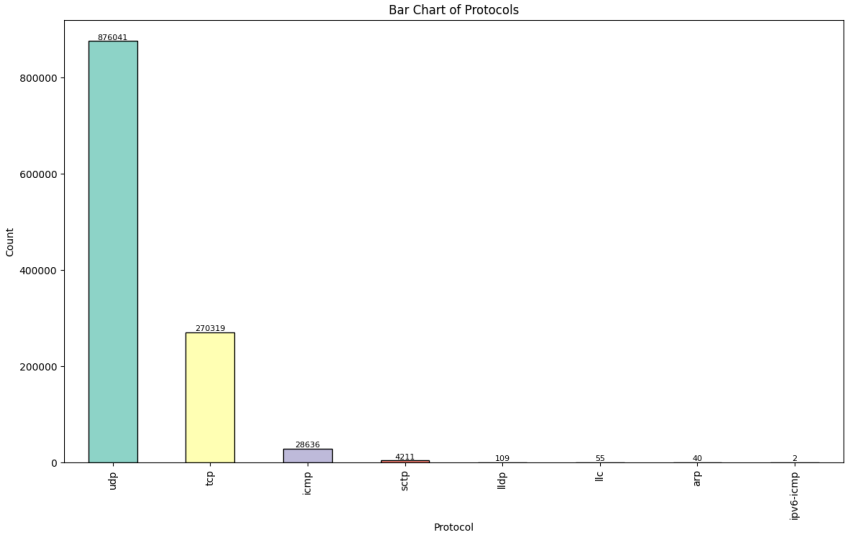
شکل ۲-۳: تعداد داده‌ها از هر کلاس



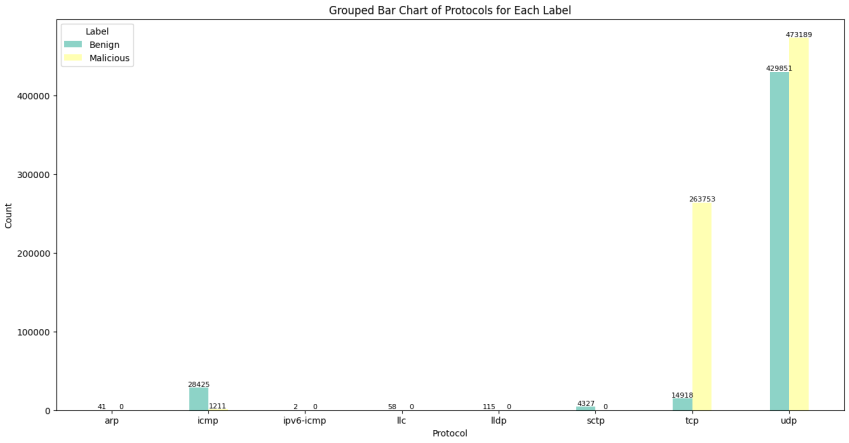
شکل ۲-۴: تعداد انواع مختلف حمله‌ها



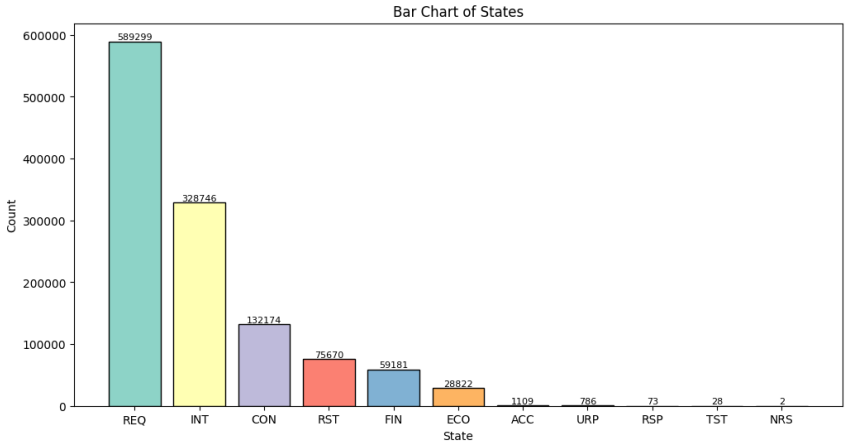
شکل ۲-۵: تعداد انواع مختلف ابزار حمله‌ها



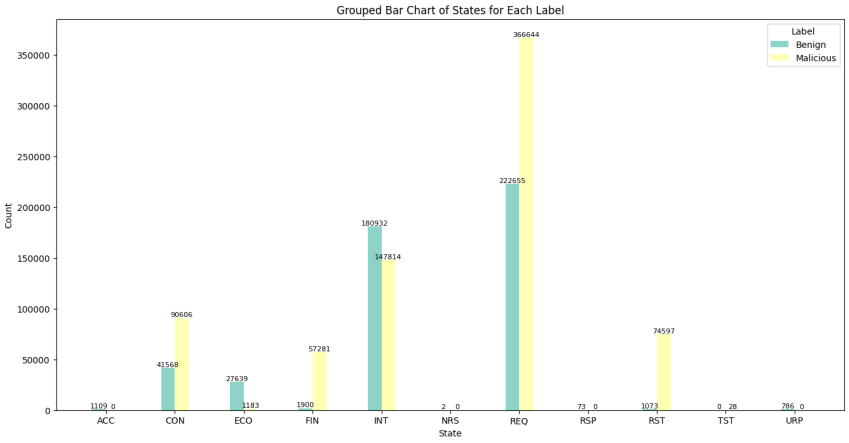
شکل ۲-۶: تعداد انواع پروتکل‌ها



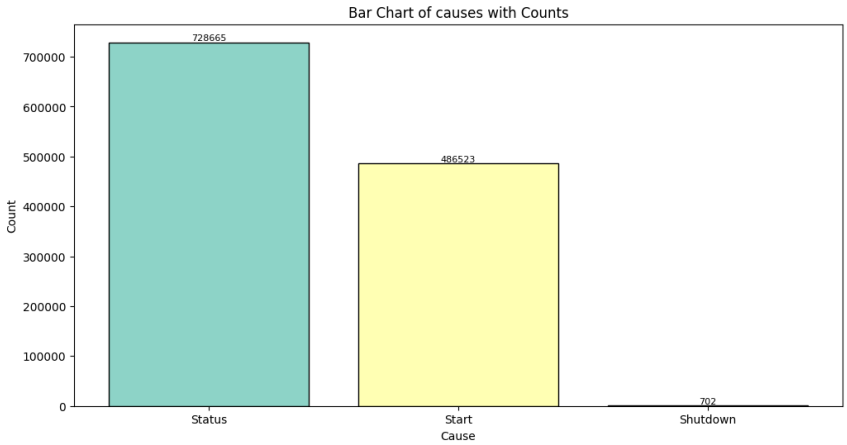
شکل ۲-۷: تعداد انواع پروتکل‌ها در هر برچسب داده



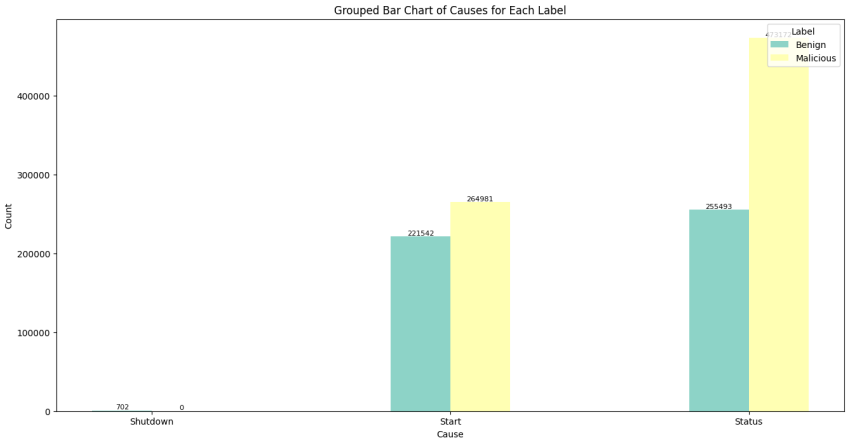
شکل ۲-۸: تعداد انواع State



شکل ۲-۹: تعداد انواع State در هر برچسب داده



شکل ۲-۱۰: تعداد انواع Cause



شکل ۲-۱۱: تعداد انواع Cause در هر برچسب داده

۲-۴ نامتوازنی در مجموعه داده‌ها

کیفیت و کمیت مجموعه داده تأثیر قابل توجهی رو عملکرد مدل‌های یادگیری ماشین و عمیق می‌گذارد. یکی از دلایل کاهش کیفیت مجموعه داده‌ها وجود نامتوازنی در آن‌ها است. نامتوازنی به معنی این است که در مجموعه داده نسبت بین کلاس‌ها یکسان نباشد و کلاس اکثریت و اقلیت موجود باشد. نامتوازنی در مجموعه داده می‌تواند باعث ایجاد جانبداری در مدل به سمت کلاس اکثریت بشود. در برخی شرایط نامتوازنی می‌تواند پیامدهای خطرناکی داشته باشد. به طور مثال، اگر تشخیص کلاس اقلیت امری حیاتی باشد و مدل جانبدارانه عمل کند و اکثر نمونه‌ها را از کلاس اکثریت پیش‌بینی کند. در شرایط نامتوازنی، معیارهای سنجش عملکرد مانند دقت نیز می‌توانند گول زننده ظاهر شوند. مثلاً در شرایط فرضی اگر مجموعه داده‌ای داشته باشیم که نسبت کلاس‌ها در آن ۹۰ به ۱۰ باشد و مدل ما همه ورودی‌هارا به کلاس اکثریت نسبت دهد، دقت ۹۰ درصد بدست می‌آید که نمی‌تواند نمایانگر ضعف مدل باشد.

مجموعه داده‌های حوزه تشخیص نفوذ نیز از این مشکل رنج می‌برند. در مجموعه داده KDD99 نسبت داده‌های بی‌خطر به حمله ۲۰ به ۸۰ است، در مجموعه داده CICIDS2017 نیز این نسبت ۸۳ به ۱۷ است. در مجموعه داده 5G-NIDD شدت نامتوازنی کمتر است و نسبت داده‌های حمله به بی‌خطر ۶۱ به ۳۹ می‌باشد. یکی از راه‌های برطرف کردن مشکل نامتوازنی، تولید داده مصنوعی‌ای است که به داده آموزشی شباهت داشته باشد.[۳۴]

فصل ۳

تولید داده مصنوعی برای متوازن کردن مجموعه داده

همان‌طور که در بخش ۲-۴ گفته شد، نامتوازنی در مجموعه داده‌ها می‌تواند بر دقت و عملکرد مدل‌های یادگیری عمیق تاثیر بگذارد. تولید داده مصنوعی می‌تواند مشکل نامتوازنی را برطرف کند. یکی از روش‌های متداول که برای این امر استفاده می‌شود روش SMOTE [۱] است. در این روش، ابتدا یک نمونه از کلاس کمینه انتخاب می‌شود و سپس نمونه‌های مصنوعی با درون‌یابی[۲] بین آن و همسایه‌های نزدیک آن، تولید می‌شوند.[۱۱] روش SMOTE مشکلاتی از قبیل همپوشانی نمونه‌ها، نویز داشتن نمونه‌ها و سختی انتخاب تعداد همسایه‌های ایده‌آل را دارد.[۲۰] مدل‌های مولد[۳] به کمک یادگیری عمیق با آموختن توزیع داده‌ها، می‌توانند داده مصنوعی تولید کنند. در این فصل به بررسی این نوع مدل‌ها و علی‌الخصوص نوع متخاصم آن‌ها می‌پردازیم.

۳-۱ انواع مدل در مسائل یادگیری عمیق

در مسائل یادگیری ماشین و یادگیری عمیق، مدل‌ها به دو دسته مدل‌های تمایزی[۴] و مدل‌های مولد تقسیم‌بندی می‌شوند که در ادامه به توضیح آن‌ها می‌پردازیم.

[1] Synthetic Minority Over-sampling Technique
[2] Interpolating
[3] Generative Models
[4] Discriminative Models

۳-۱-۱ مدل‌های تمایزی

مدل‌های تمایزی، مدل‌هایی هستند که توزیع شرطی $P(y|x)$ را یاد می‌گیرند. x نمایانگر ویژگی‌های یک نمونه و y برچسب متناظر هر نمونه است. این مدل‌ها مرزهای تصمیم را از طریق داده های مشاهده شده، مانند پاس/شکست، برد/باخت، زنده/مرده یا سالم/بیمار از هم تشخیص می دهند.[۲] از انواع این مدل ها می‌توان به مدل‌های زیر اشاره کرد.

- دسته‌بند خطی[۱]
- وایازش لجستیک[۲]
- درخت تصمیم[۳]

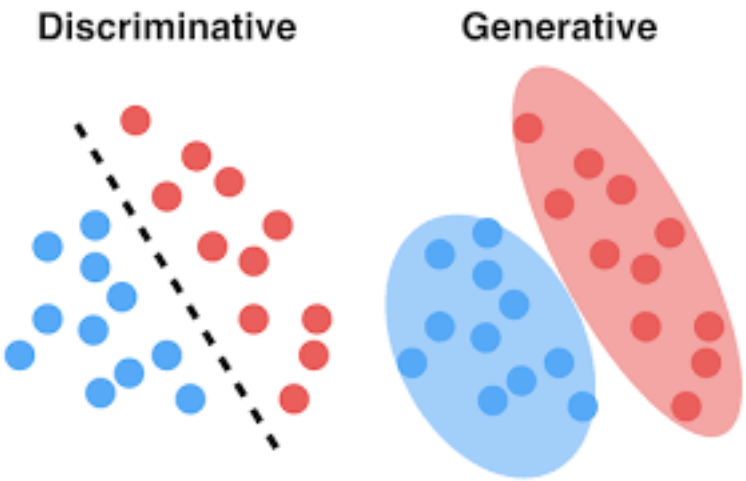
۳-۱-۲ مدل‌های مولد

این مدل‌ها برخلاف مدل‌های تمایزی توزیع توأم $P(x, y)$ را یاد می‌گیرند و با استفاده از توزیع آموخته شده توانایی تولید نمونه جدید را دارند.[۳] با پیشرفت یادگیری عمیق زمینه برای پیشرفت این مدل‌ها به وجود آمد و از نمونه‌های آن در حوزه پردازش متن می‌توان به ChatGPT اشاره کرد. از انواع مدل‌های مولد می‌توان به مدل‌های زیر اشاره کرد.

- شبکه مولد متخاصم
- خودرمزگذار متغیر[۴]
- مدل‌های انتشاری [۵]

در ادامه به توضیح عملکرد شبکه مولد متخاصم می‌پردازیم.

[1] Linear Classifier
[2] Logistic Regression
[3] Decision Tree
[4] Variational Autoencoder
[5] Diffusion Model



شکل ۳-۱: تفاوت مدل‌های تمایزی و مولد [۱]

۳-۲ شبکه‌های مولد متخاصم

در سال‌های اخیر، حوزه هوش مصنوعی، به ویژه در حوزه مدل‌های مولد، شاهد پیشرفت‌های چشمگیری بوده است. در این میان، شبکه‌های متخاصم مولد به عنوان یک الگوی انقلابی ظهور کرده‌اند که رویکردی قدرتمند برای تولید داده‌های واقع‌گرایانه از طریق آموزش خصمانه[1] ارائه می‌دهد. شبکه‌های مولد متخاصم در سال ۲۰۱۴ معرفی شدند و از آن زمان تاکنون این امکان را به محققان داده‌اند که داده‌های مصنوعی که ویژگی های نمونه های دنیای واقعی را منعکس می کند، تولید کنند.[۱۶]

تطبیق‌پذیری شبکه‌های مولد متخاصم در تولید داده در دامنه‌های مختلف گسترش یافته است. این مدل‌ها در راستای حل مسائل مرتبط با کمبود داده‌های آموزشی یا ناهمواری‌های توازن مجموعه داده، نقش حیاتی و اساسی دارند. این تکنولوژی‌ها به عنوان ابزاری کارآمد واجد اهمیت به‌شمار می‌آیند، زیرا می‌توانند با ارائه راهکارهای نوآورانه و هوشمندانه، مشکلات مرتبط با نقصان یا عدم توازن در داده‌های آموزشی را بهبود بخشند. این مدل‌ها علاوه بر افزایش حجم داده‌های آموزشی، بهبود کیفیت و تنوع این داده‌ها را نیز هدف قرار می‌دهند. به طور کلی، مدل‌های مورد استفاده در این زمینه‌ها نقش فعّالی را در توسعه و بهبود عملکرد الگوریتم‌ها و سیستم‌های هوش مصنوعی بازی می‌کنند.

در ادامه به معماری این مدل‌ها، نحوه آموزش، کاربرد و روش استفاده از آن‌ها در این پروژه اشاره می‌کنیم.

[1] Adversarial training

۳-۲-۱ معماری شبکه‌های مولد متخاصم

نقطه قوت شبکه‌های مولد متخاصم در معماری نوآورانه و در عین حال پویا آن‌ها نهفته است، که باعث ایجاد یک تعامل مداوم بین دو شبکه عصبی شده است. این دو شبکه عصبی عبارتند از مولد[۱] و تفکیک‌کننده[۲].[۱۶] درک این معماری و شبکه‌ها برای درک ماهیت چگونگی تولید داده‌های واقع‌گرایانه حائز اهمیت است.

۳-۲-۲ مولد در شبکه مولد متخاصم

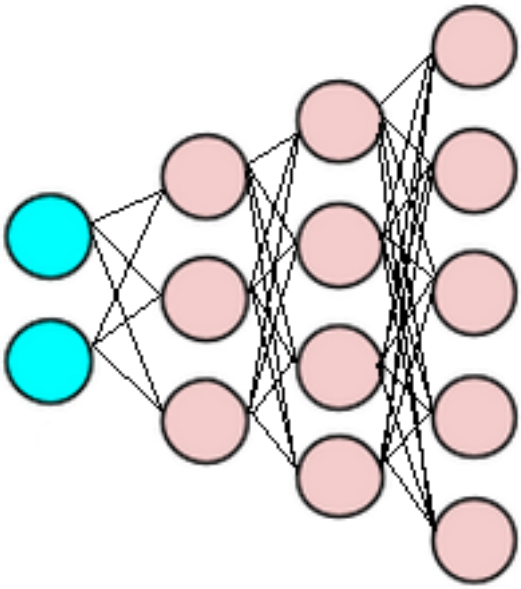
در قلب معماری شبکه مولد متخاصم، شبکه مولد وجود دارد که برای ایجاد داده مصنوعی طراحی شده است. این شبکه که یک شبکه عصبی است، نویزی را به عنوان ورودی می‌گیرد و وظیفه دارد آن را به خروجی تبدیل کند. هدف شبکه مولد این است که خروجی‌ای که تولید می‌کند با معنا باشد و از داده‌های واقعی قابل تمایز نباشند. در طی فرآیند یادگیری، این شبکه توانایی خود برای تبدیل نویز ورودی به به داده‌های واقع‌گرایانه، بهبود می‌بخشد. معماری این شبکه می‌تواند بر اساس ماهیت داده‌ای که برای تولید آن طراحی شده‌است متفاوت باشد و می‌تواند از لایه‌های خطی[۳]، لایه‌های همگشتی[۴] و یا لایه‌های بازگشتی[۵] تشکیل شوند. آزمایش‌ها نشان می‌دهند که توزیع نویز چندان اهمیتی ندارد، بنابراین می‌توانیم چیزی را انتخاب کنیم که نمونه‌برداری از آن آسان باشد، مانند توزیع یکنواخت[۶].[۶]

در تصویر ۳-۲ تصویر ساده‌ای از ساختار مولد قابل مشاهده است که نویزی با ابعاد ۲ را تبدیل به خروجی با ابعاد ۵ می‌کند.

۳-۲-۳ تفکیک‌کننده در شبکه مولد متخاصم

تفکیک‌کننده در شبکه مولد متخاصم نقش حریف مولد را دارد. هدف آن‌ها این است که داده‌های اصلی را از داده‌های مصنوعی تولید شده توسط مولد تفکیک کند. در طی فرآیند آموزش خصمانه، تفکیک‌کننده در دسته‌بندی نمونه‌ها ماهر می‌شود و مولد را به بهبود مستمر توانایی خود در ایجاد داده‌های واقع‌گرایانه سوق می‌دهد. مشابه مولد، تفکیک‌کننده نیز یک شبکه عصبی است و طراحی شده که ویژگی‌های نمونه‌ها را بررسی

[1] Generator
[2] Discriminator
[3] Linear Layer
[4] Convolutional Layer
[5] Recurrent Layer
[6] Uniform distribution



شکل ۳-۲: ساختار یک شبکه مولد ساده

کند و تشخیص دهد که داده واقعی و یا مصنوعی است. خروجی تفکیک‌کننده یک عدد بین ۰ و ۱ است و نشان دهنده این است که با چه احتمالی داده ورودی به توزیع داده اصلی تعلق دارد.[۱۶]

۳-۲-۴ فرآیند آموزش در شبکه مولد متخاصم

قدرت شبکه مولد متخاصم در مرحله آموزش خصمانه آشکار می‌شود. مولد به دنبال تولید داده‌هایی است که از نمونه‌های واقعی قابل تشخیص نیستند، در حالی که تفکیک‌کننده تلاش می‌کند تا در تفکیک ماهرتر شود. این رقابت باعث می‌شود که مولد و تفکیک‌کننده بهینه شوند و بتوانند وظیفه خود یعنی تولید داده مصنوعی و تشخیص داده مصنوعی از داده واقعی را به نحو احسن انجام دهند.

تابع ضرر شبکه تفکیک‌کننده به گونه‌ای باید باشد که که اگر داده واقعی به عنوان داده مصنوعی و یا اگر داده مصنوعی به عنوان داده واقعی پیش‌بینی شود، جریمه بشود. تابع ضرر تفکیک‌کننده به صورت زیر تعریف می‌شود.

$$\frac{۱}{m}\sum_{i=۱}^{m}\left[\log D\left(x^{(i)}\right)+\log\left(۱-D(G(z^{(i)}))\right)\right] \qquad (۱-۳)$$

در تابع (۳-۱) $D(x^{(i)})$ خروجی تفکیک‌کننده به ازای هر نمونه از ورودی اصلی، $z^{(i)}$ یک نمونه از توزیع نویز، $G(z^{(i)})$ نمونه مصنوعی تولید شده توسط مولد برای نویز ورودی و $D(G(z^{(i)}))$ خروجی تفکیک‌کننده برای نمونه مصنوعی تولید شده می‌باشد. هدف تفکیک‌کننده این است که این تابع را به حداکثر برساند که خروجی آن برای نمونه‌های واقعی نزدیک به ۱ و برای نمونه‌های مصنوعی، نزدیک به ۰ باشد.

شبکه مولد باید سعی بر این کند که تفکیک‌کننده را گول بزند و نمونه‌هایی تولید کند که از نمونه‌های واقعی تفکیک‌پذیر نباشند. تابع ضرر آن به صورت زیر تعریف می‌شود.

$$\frac{1}{m}\sum_{i=1}^{m}\left[\log\left(1 - D(G(z^{(i)}))\right)\right] \qquad (۳-۲)$$

هدف مولد این است که تابع (۳-۲) را حداقل کند تا خروجی تفکیک‌کننده برای داده‌های تولید شده از روی نویز نزدیک به ۱ باشد. مراحل آموزش شبکه مولد متخاصم به شرح زیر می‌باشد.

۱. در ابتدای آموزش، مولد و تفکیک‌کننده با وزن‌های تصادفی تعریف می‌شوند.

۲. m نمونه نویز و m نمونه داده اصلی انتخاب می‌شود.

۳. خروجی مولد برای نمونه نویزها محاسبه می‌شود.

۴. خروجی تفکیک‌کننده برای نمونه‌داده‌های اصلی و برای نمونه‌های تولید شده توسط مولد محاسبه می‌شوند.

۵. با اجرا الگوریتم گرادیان افزایشی[1] روی تابع (۳-۱)، وزن‌های تفکیک‌کننده بروزرسانی می‌شوند.

۶. m نمونه نویز انتخاب می‌شود.

۷. خروجی مولد برای نمونه نویزها محاسبه می‌شود.

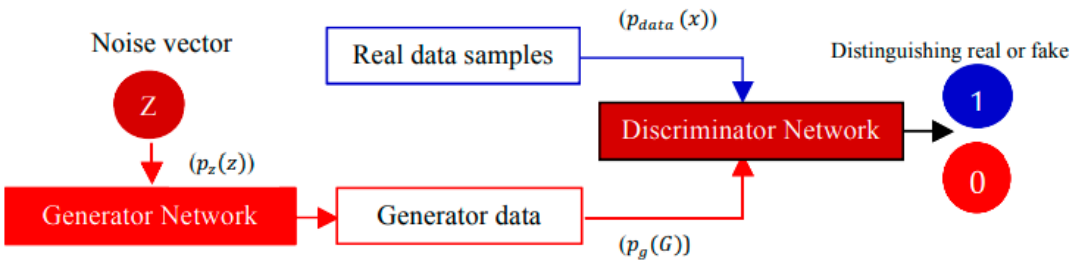
۸. خروجی تفکیک‌کننده برای نمونه‌های تولید شده توسط مولد محاسبه می‌شود.

۹. با اجرا الگوریتم گرادیان کاهشی[2] روی تابع (۳-۲)، وزن‌های مولد بروزرسانی می‌شوند.

[1] Gradient Ascent
[2] Gradient Descent

۱۰. مراحل ۲ تا ۹ برای هر دوره[1] از آموزش تکرار می‌شوند.

در تصویر ۳-۳ معماری این نوع شبکه قابل مشاهده می‌باشد.



شکل ۳-۳: معماری شبکه مولد متخاصم[۱۹]

۳-۲-۵ محدودیت شبکه‌های مولد متخاصم

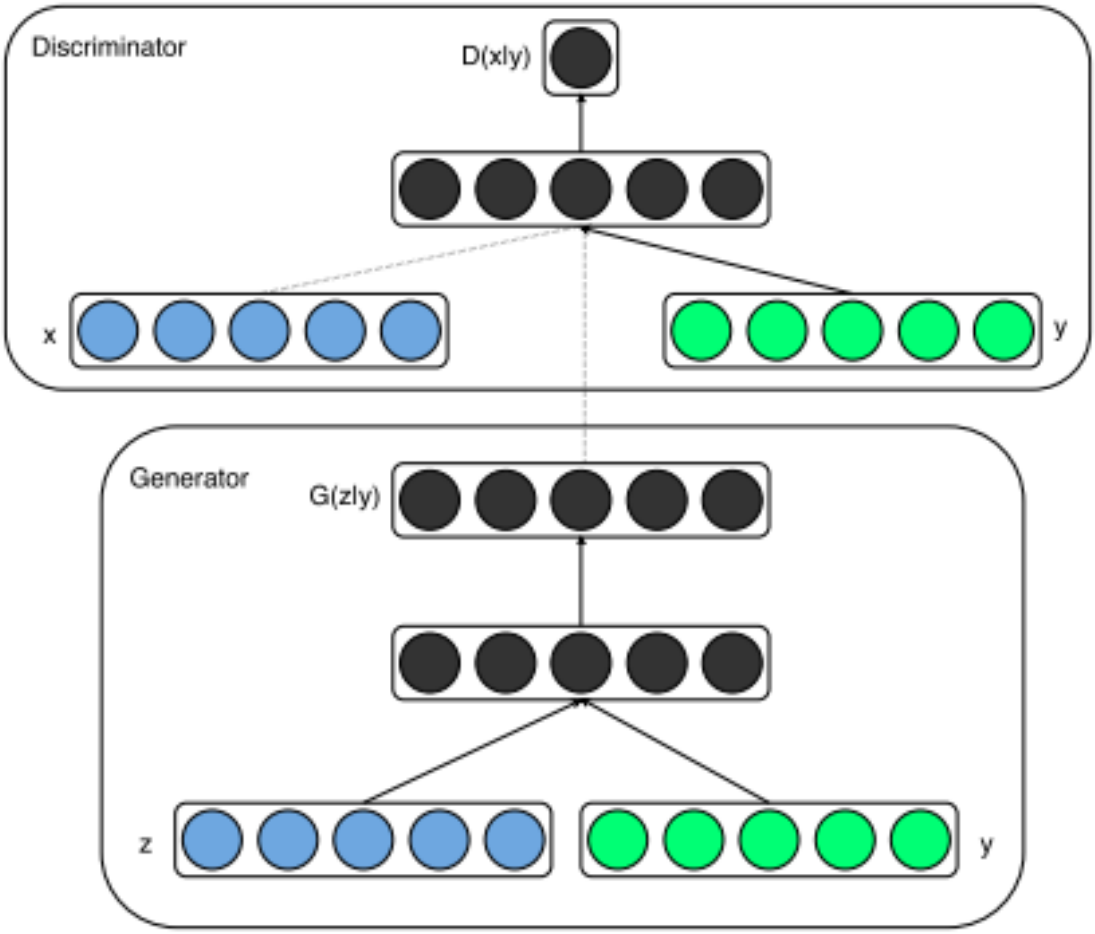
با وجود مزیت‌های فراوان، یکی از محدودیت‌های شبکه مولد متخاصم این است که نظارتی روی داده تولیدی وجود ندارد. به طور مثال این امکان وجود ندارد که بخواهیم فقط از یک کلاس موجود در مجموعه داده، نمونه مصنوعی تولید کنیم و نمونه‌های تولید شده می‌توانند با نسبت‌های تصادفی به کلاس‌ها تعلق داشته باشند.[۲۵] به طور مثال اگر ۲۰۰ نمونه از کلاس اقلیت بخواهیم تولید کنیم و نسبت دو کلاس در داده‌های تولید شده ۸۰ به ۲۰ باشد، باید ۱۰۰۰ نمونه تولید کنیم که از نظر محاسباتی بهینه نمی‌باشد. برای برطرف کردن این مشکل، شبکه‌های مولد متخاصم شرطی معرفی شده‌اند که در ادامه به بررسی آن‌ها می‌پردازیم.

۳-۲-۶ شبکه مولد متخاصم شرطی

همان‌طور که در بخش ۳-۲-۵ گفته شد، یکی از محدودیت‌های شبکه مولد متخاصم، این است که نظارتی روی داده تولید شده وجود ندارد. راه‌حل پیشنهادی برای این مشکل، اضافه کردن شروط به ورودی شبکه مولد متخاصم می‌باشد.[۲۵] با این روش می‌توان بر داده تولید شده نظارت کرد. شروط اضافه شده می‌تواند برچسب داده، خصوصیات داده خروجی و یا هر چیزی که رو داده تاثیر بگذارد، باشد. در این پروژه شرط ورودی، کلاس داده‌ها یعنی حمله یا بی‌خطر می‌باشد.

[1] Epoch

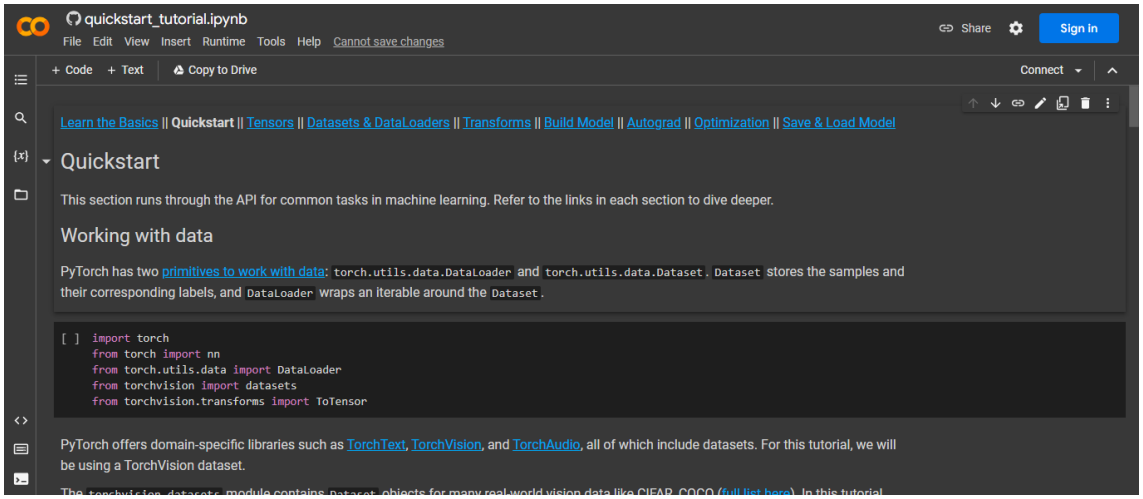
نحوه آموزش این شبکه نسبت به حالت معمولی آن، شامل تغییرات اندکی می‌شود. ورودی مولد و تفکیک‌کننده حالت شرطی می‌گیرد یعنی در فرمول (۳-۱) و (۳-۲) $D(x^{(i)})$ به $D(x^{(i)}|y^{(i)})$ و $G(z^{(i)})$ به $D(z^{(i)}|y^{(i)})$ تبدیل می‌شود که $y^{(i)}$ شرط اضافه شده به ورودی می‌باشد. در تصویر ۳-۴ این تغییرات مشهود می‌باشند.



شکل ۳-۴: تغییرات ورودی در شبکه مولد متخاصم شرطی[۲۵]

۳-۲-۷ پیاده‌سازی شبکه مولد متخاصم شرطی

در این پروژه برای پیاده‌سازی شبکه مولد متخاصم شرطی از کتاب‌خانه sdv استفاده شده است که مستندات آن در این آدرس موجود است.



شکل ۳-۵: محیط کاربری Google Colab

برای اجرا گرفتن از ابزار Google Colab استفاده شده که کارت گرافیک T4 را به صورت رایگان ولی با محدودیت زمانی در اختیار عموم قرار می‌دهد. محیط کاربری این ابزار در تصویر ۳-۵ قابل مشاهده است. این ابزار که در بین توسعه‌دهنده‌ها و محققان حوزه یادگیری عمیق محبوب است، این امکان را به وجود می‌آورد که بدون در اختیار داشتن سخت‌افزارهای گران و هزینه‌بر، بتوان آزمایشات و برنامه‌ها به زبان R و Python را اجرا کرد.

ابرپارامتر[1]های این شبکه عبارت است از

۱. در تفکیک‌کننده و مولد ۲ لایه مخفی وجود دارد که هر کدام ۲۵۶ نورون دارند.

۲. به منظور جلوگیری از بیش‌برازش [2]، از روش حذف تصادفی [3] با احتمال ۰.۵ استفاده شده است.

۳. به علت محدودیت سخت‌افزاری، امکان پردازش روی همه داده‌ها به صورت همزمان وجود ندارد و آن‌ها را به دسته‌های ۵۰۰تایی تقسیم‌بندی کردیم.

۴. تعداد دوره‌های[4] آموزش برابر ۲۰ قرار داده شده است.

[1] Hyperparameter
[2] Overfitting
[3] Dropout
[4] Epoch

۵. از الگوریتم بهینه‌سازی Adam[۲۱] برای بروزرسانی پارامترهای شبکه استفاده می‌شود.

۶. نرخ یادگیری در الگوریتم Adam برای هر دو شبکه مولد و تفکیک‌کننده برابر ۲×۱۰^{-۴} در نظر گرفته می‌شود.

همانطور که در بخش ۲ گفته شد، بعد از تقسیم مجموعه داده به دو مجموعه آموزش و آزمایشی، تعداد ۱۱۷۹۴۱۳ نمونه آموزشی در دسترس است که از این تعداد، ۷۱۶۰۶۰ تا برچسب حمله و ۴۶۳۳۵۳ تا برچسب بی‌خطر دارند. برای تولید داده مصنوعی نیز همین نسبت و تعداد را در نظر گرفتیم.

۳-۲-۸ ارزیابی داده‌های مصنوعی تولید شده

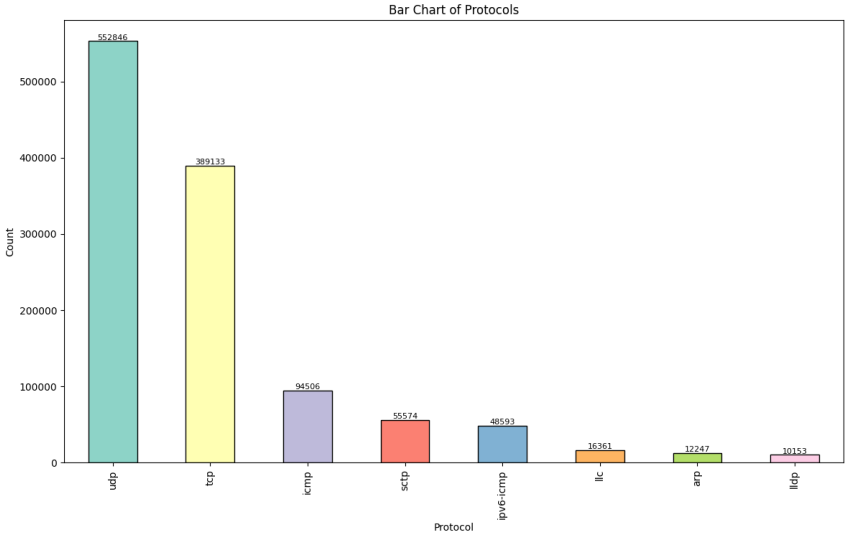
شبکه‌های مولد متخاصم در ابتدا در حوزه پردازش تصویر، برای تولید عکس به وجود آمدند و برای ارزیابی عملکرد آن‌ها روش Inception score[۱] معرفی شد که با استفاده از یک مدل یادگیری عمیق که از قبل روی یک مجموعه داده بزرگ آموزش داده شده بود، عملکرد مدل و مناسب بودن داده‌های مصنوعی را بررسی می‌کردند.[۳۲] برای مجموعه داده ما که جدولی[۲] است، این روش ممکن نیست زیرا ویژگی‌ها در مجموعه داده‌های جدولی یکسان نیست و مدلی از پیش آموخته[۳] وجود ندارد.

با ایده گرفتن از این روش، بر روی مجموعه داده آموزش یک شبکه عصبی دسته‌بندی ساده با یک لایه مخفی آموزش می‌دهیم و عملکرد همین مدل را روی داده‌های مصنوعی ارزیابی می‌کنیم. دقت این مدل بر روی داده آموزش ۹۶ درصد بدست آمد و روی داده‌های مصنوعی به دقت مناسب ۹۲ درصد رسید. این نشان‌دهنده این است که داده تولید شده از کیفیت بالایی برخوردار است و الگوهای اساسی موجود در مجموعه داده اصلی را حفظ می‌کند.هم‌ترازی نزدیک در عملکرد بین مدل بر روی داده‌های واقعی و مصنوعی نشان می‌دهد که داده‌های تولید شده برای کار مورد نظر ما مناسب هستند، و پتانسیل این را دارند که مجموعه داده را تقویت کنند.

[۱] اسم آن از روی مدل inception-v3 گرفته شده است.

[2] Tabular

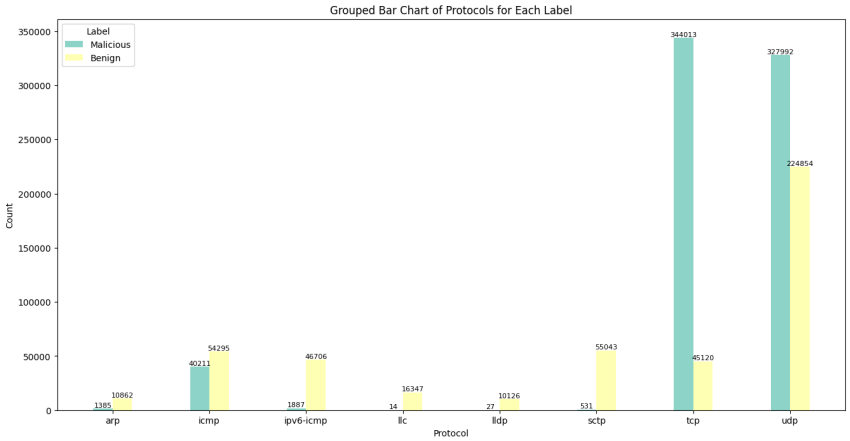
[3] Pre-trained



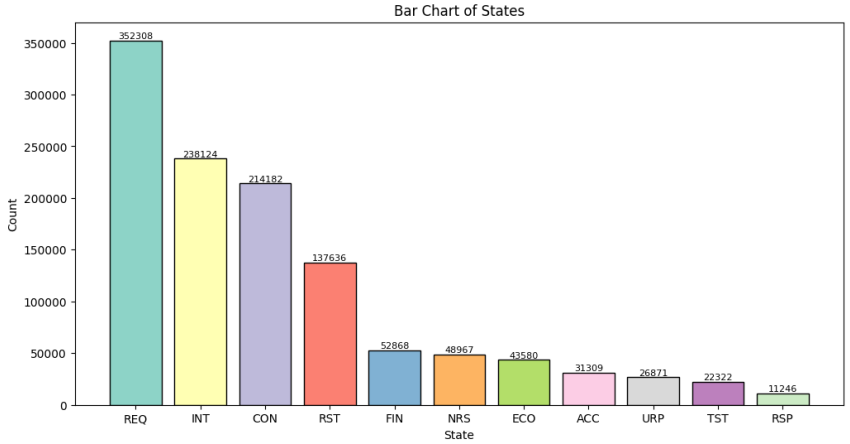
شکل ۳-۶: تعداد انواع پروتکل‌ها در مجموعه داده مصنوعی

۳-۲-۹ گزارشی از داده‌های مصنوعی تولید شده

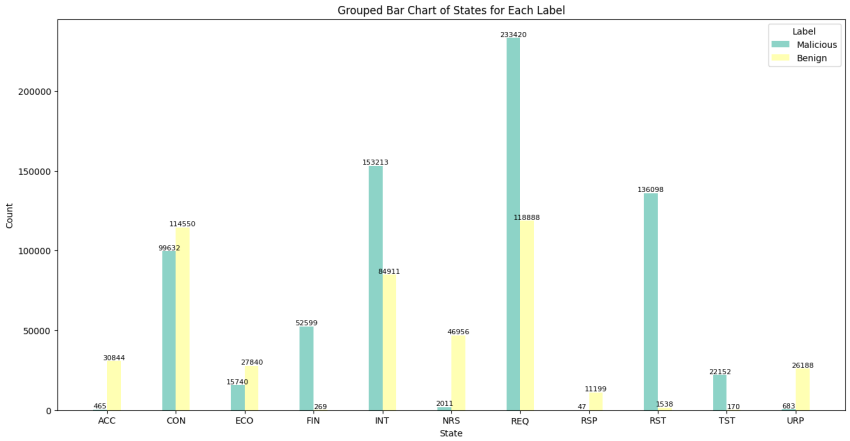
همانند بخش ۳-۲ در این قسمت نیز گزارشی از مجموعه داده مصنوعی تولید شده آماده می‌کنیم. این گزارشات در شکل‌های ۳-۶ تا ۳-۱۱ نمایش داده شده‌اند.



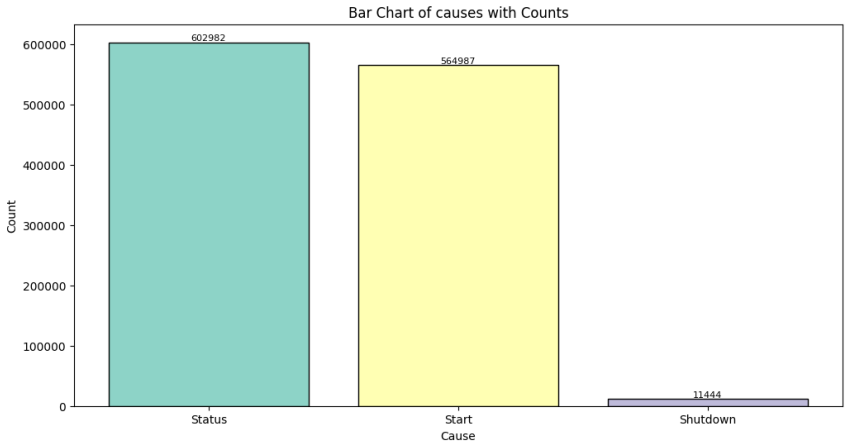
شکل ۳-۷: تعداد انواع پروتکل‌ها در هر برچسب داده در مجموعه داده مصنوعی



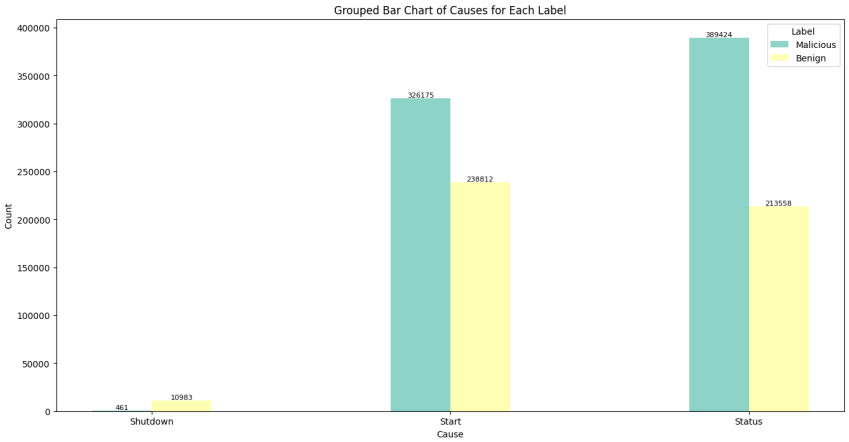
شکل ۳-۸: تعداد انواع State در مجموعه داده مصنوعی



شکل ۳-۹: تعداد انواع State در هر برچسب داده در مجموعه داده مصنوعی



شکل ۳-۱۰: تعداد انواع Cause در مجموعه داده مصنوعی



شکل ۳-۱۱: تعداد انواع Cause در هر برچسب داده در مجموعه داده مصنوعی

فصل ۴

دسته‌بندی داده‌ها

برای توسعه یک سامانه تشخیص نفوذ شبکه، احتیاج به این است که ترافیک‌های شبکه مورد پردازش قرار بگیرند و با توجه به ویژگی‌ها و الگوهای موجود در آن‌ها به دو دسته حمله و بی‌خطر دسته‌بندی شوند. در این فصل ابتدا به پژوهش‌های پیشین در زمینه دسته‌بندی در سامانه‌های تشخیص نفوذ می‌پردازیم، سپس شاخص‌های ارزیابی در یادگیری عمیق معرفی می‌شوند، مدل مورد استفاده ما برای دسته‌بندی معرفی می‌شود و در پایان نتایج بدست آمده ارائه می‌شوند.

۴-۱ پژوهش‌های پیشین

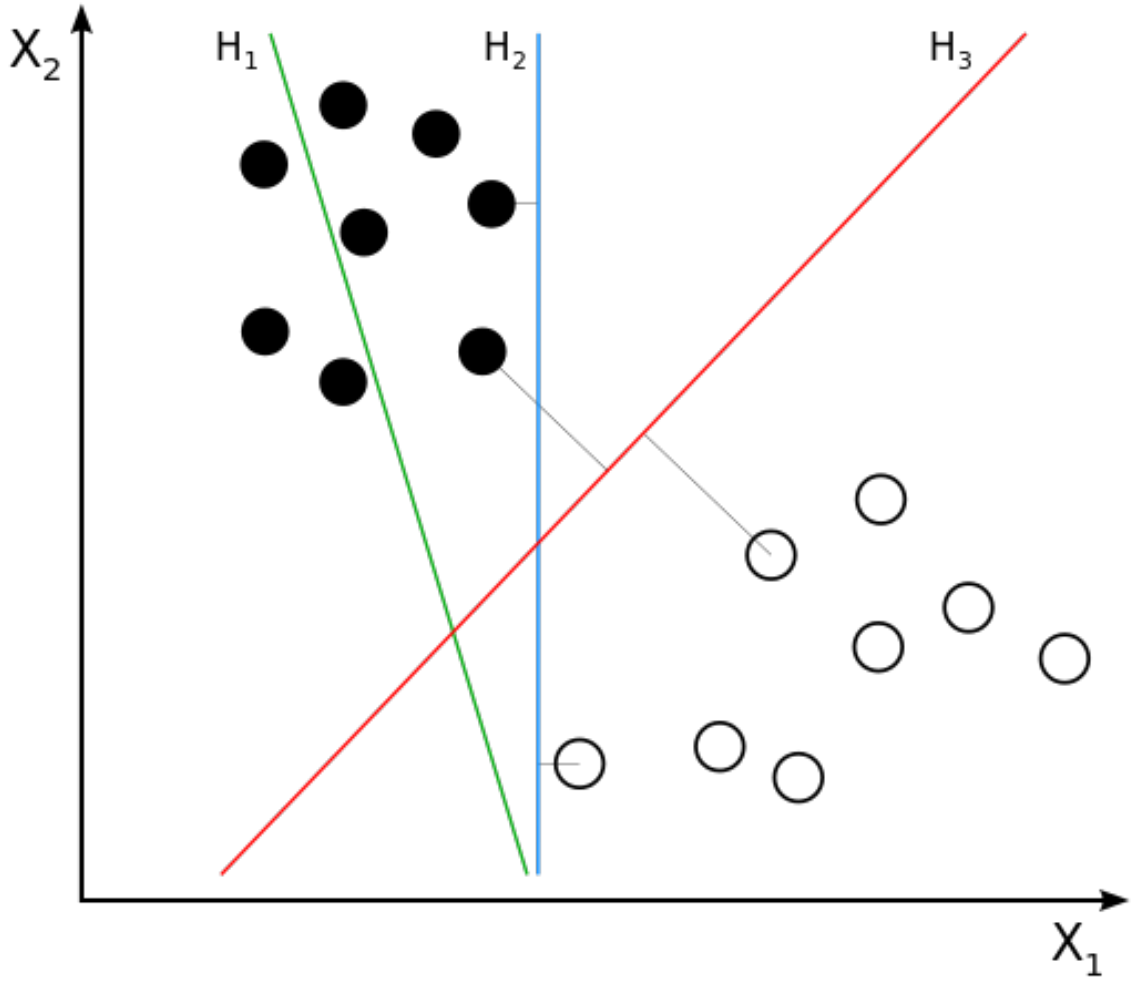
روش‌های یادگیری ماشین و یادگیری عمیق برحسب نوع داده به دو روش یادگیری نظارت‌شده[۱] و یادگیری بدون نظارت[۲] تقسیم می‌شوند. در این بخش به مقایسه این دو روش می‌پردازیم.

۴-۱-۱ یادگیری نظارت‌شده

در این روش یادگیری، مجموعه داده دارای برچسب می‌باشد و مدل با بررسی ویژگی‌های هر نمونه ورودی، خروجی را پیش‌بینی می‌کند. از انواع این روش‌ها می‌توان به نمونه‌های زیر اشاره کرد.

[1] Supervised Learning
[2] Unsupervised Learning

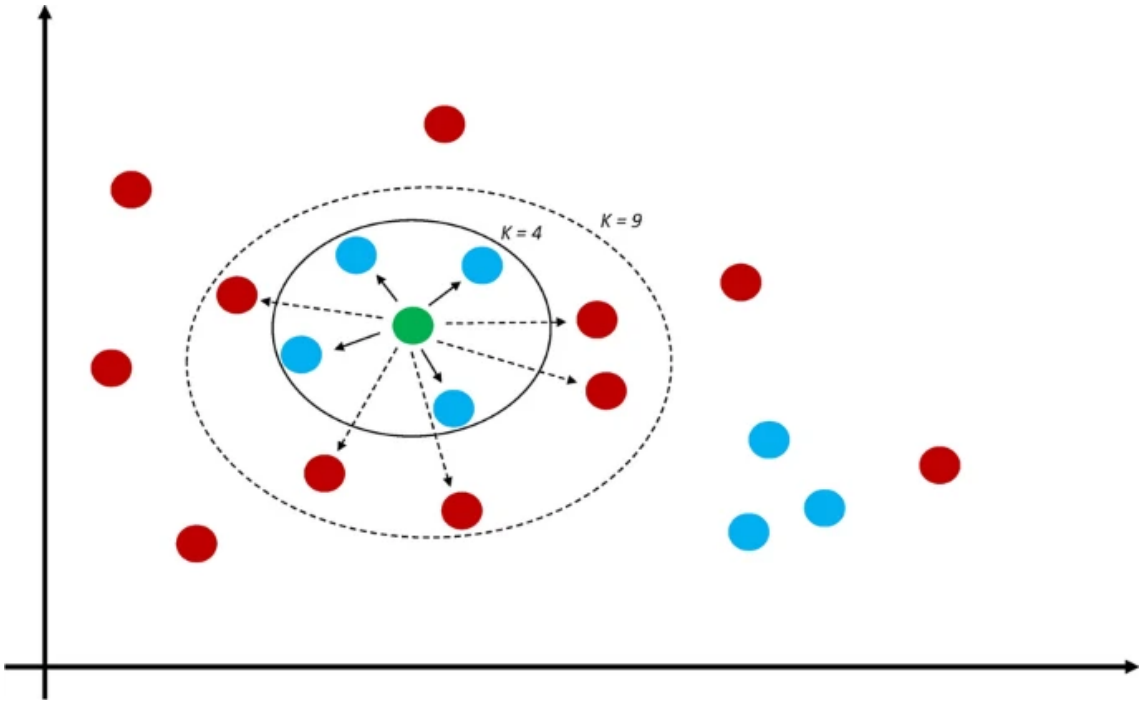
۱. ماشین بردار پشتیبان[1]: ماشین بردار پشتیبان، با یافتن ابرصفحه[2] بهینه، داده‌های مربوط به کلاس‌های مختلف را در فضایی با ابعاد بالا تفکیک می‌کند.در تقسیم خطی داده‌ها سعی می‌کنیم ابرصفحه ای را انتخاب شود که حاشیه اطمینان بیشتری داشته باشد.[۲۳، ۵] در تصویر ۴-۱ ابرصفحه H_3 از بقیه بهتر است، چون حاشیه اطمینان بیشتری دارد. ماشین‌های بردار پشتیان توانایی این را دارند که با استفاده از توابع هسته[3] مختلف، داده‌هایی که خطی یا غیرخطی جداپذیر هستند را تفکیک کنند.[۳۹]



شکل ۴-۱: ابرصفحه‌های متفاوت برای داده‌ها[۵]

[1]Support Vector Machine
[2]Hyperplane
[3]Kernel Function

۲. K نزدیک‌ترین همسایه[1]: این الگوریتم، یک روش ساده‌ی نظارت‌شده است که نمونه‌های ورودی را بر مبنای مفهوم مشابهت ویژگی‌ها دسته بندی می کند. نمونه‌ای از مجموعه‌داده انتخاب می‌شود و برچسب آن با توجه به k نمونه نزدیک آن انتخاب می‌شود. انتخاب k و همچنین نحوه محاسبه فاصله در عملکرد الگوریتم اثر می‌گذارد.[۷] فاصله اقلیدسی یک انتخاب محبوب برای محاسبه فاصله می‌باشد. در تصویر ۴-۲ در حالتی که k برابر ۴ است داده از کلاس آبی پیش‌بینی می‌شود ولی در حالتی که برابر ۹ است، از کلاس قرمز پیش‌بینی می‌شود.

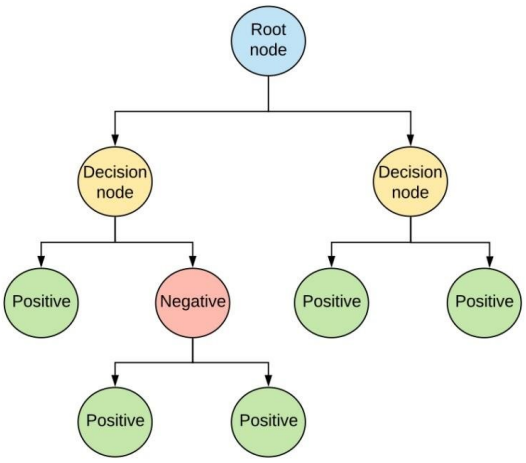


شکل ۴-۲: تأثیر انتخاب k روی پیش‌بینی برچسب داده[۷]

۳. درخت تصمیم[2]: یکی از روش‌های بنیادین در یادگیری نظارت‌شده، درخت تصمیم است. در این روش، مدل‌ها بر اساس ویژگی‌های نمونه ورودی، یک سری تصمیمات منطقی می‌گیرند و برچسب داده را پیش‌بینی می‌کنند. ساختار درخت با یک گره ریشه شروع می‌شود که کل مجموعه داده را نشان می‌دهد. در هر گره داخلی، یک تصمیم بر اساس یک ویژگی گرفته می‌شود که به گره‌های بعدی

[1] K-Nearest Neighbors
[2] Decision Tree

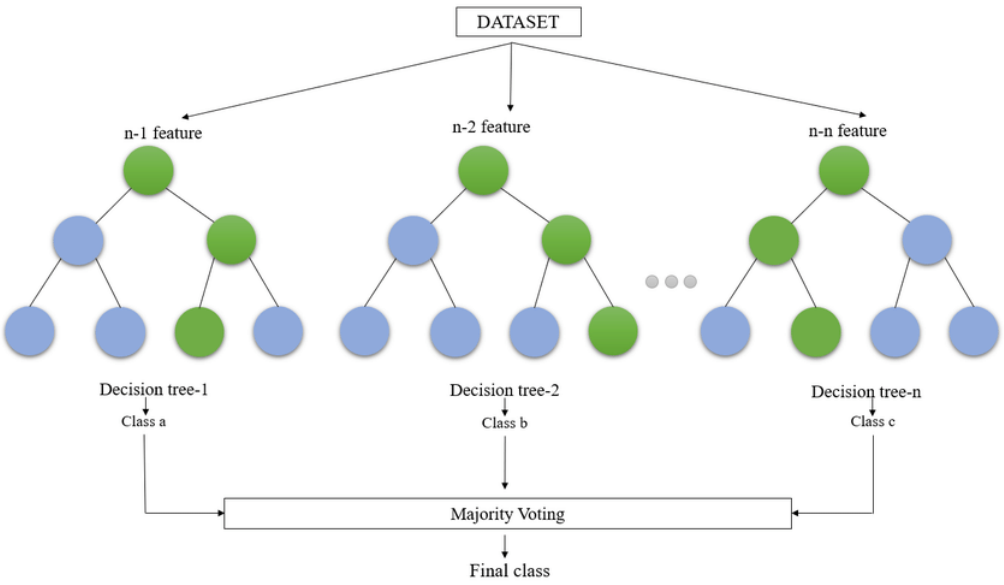
منشعب می‌شود که نتایج ممکن را نشان می‌دهند.[۱۳] این روند تا رسیدن به برگ‌ها ادامه می‌یابد که نشان‌دهنده خروجی درخت یا همان برچسب پیش‌بینی شده می‌باشد. ساختار درخت تصمیم در تصویر ۴-۳ قابل مشاهده است.



شکل ۴-۳: ساختار درخت تصمیم[۳۵]

۴. جنگل تصادفی[1]: جنگل تصادفی یک روش یادگیری تجمیعی[2] است که به نوعی توسعه‌یافته درخت تصمیم به شمار می‌رود. در یادگیری تجمیعی، چندین مدل به منظور کاهش خطا و بهبود عملکرد، مورد استفاده قرار می‌گیرند. جنگل تصادفی از چندین درخت تصمیم تشکیل شده است و نمونه‌ها به همه درخت‌ها به عنوان ورودی داده می‌شوند و خروجی درخت‌ها برای آن‌ها بدست می‌آید. کلاسی که اکثر درخت‌ها آن را پیش‌بینی کنند به عنوان خروجی نهایی در نظر گرفته می‌شود.[۱۰] در تصویر ۴-۴ ساختار جنگل تصادفی قابل مشاهده است.

[1] Random Forest
[2] Ensemble Learning



شکل ۴-۴: ساختار جنگل تصادفی که از ۳ درخت تصمیم تشکیل شده است.

۵. بیز ساده[1]: بیز ساده یک مدل یادگیری ماشین احتمالاتی بر اساس قضیه بیز می‌باشد.این مدل بر این فرض متکی است که ویژگی‌ها نسبت به هم مستقل باشند. یعنی به ازای هر ویژگی داشته باشیم $P(x_1, x_2, ..., x_n|C) = P(x_1|C)P(x_2|C)...P(x_n|C)$. در حالی که بیز ساده یک الگوریتم قدرتمند است، اتکای آن به فرض استقلال ویژگی‌ها می‌تواند محدودیتی در سناریوهایی که ویژگی‌ها همبستگی دارند، به وجود آورد. برچسب نمونه‌ها در مدل بیز ساده بر اساس فرمول ۴-۱ پیش‌بینی می‌شود.

$$P(y|x_1, x_2, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, x_2, ..., x_n)} \qquad (۴-۱)$$

۶. شبکه عصبی[2]: دستگاه عصبی انسان، الهام بخش این روش نظارت‌شده‌ی یادگیری ماشین است که شامل نورون‌های عصبی و اتصالات بین آن‌ها می‌باشد. نورون‌ها در شبکه عصبی به صورت یک لایه‌ی ورودی، چندین لایه مخفی و یک لایه خروجی سامان‌دهی می‌شوند. اتصالات بین نورون‌ها شامل وزن می‌باشد و خروجی هر نورون از جمع وزن‌دار خروجی‌های لایه قبل که یک تابع فعال‌سازی[3] روی آن

[1] Naive Bayes
[2] Neural Network
[3] Activation Function

اعمال می‌شود، محاسبه می‌شود. فرمول ۴-۲ نحوه محاسبه خروجی نورون‌ها را نشان می‌دهد.

$$o = f(w_۱x_۱ + w_۲x_۲ + \ldots + w_nx_n + b) \qquad (۴-۲)$$

علاوه بر وزن‌ها، هر نورون شامل مقدار ثابت انحراف[1] می‌باشد که در فرمول ۴-۲ با b نشان داده شده است.

توابع فعال‌سازی باید توابعی غیر خطی باشند که به مدل توانایی تشخیص الگوهای غیر خطی را بدهد. همچنین از این توابع برای کنترل بازه خروجی نورون‌ها استفاده می‌شود. به طور مثال اگر خروجی شبکه، احتمال باشد، تابع فعال‌سازی اعمال شده روی لایه آخر، باید مقدار بین ۰ و ۱ برگرداند.توابع فعال‌سازی که اغلب از آن‌ها استفاده می‌شوند عبارتند از:

- تابع سیگموئید:

$$\sigma(z) = \frac{۱}{۱ + e^{-z}} \qquad (۴-۳)$$

- تابع یکسوساز:

$$ReLU(x) = \begin{cases} x & x > ۰ \\ ۰ & \text{otherwise} \end{cases} \qquad (۴-۴)$$

- تابع تانژانت هیپربولیک:

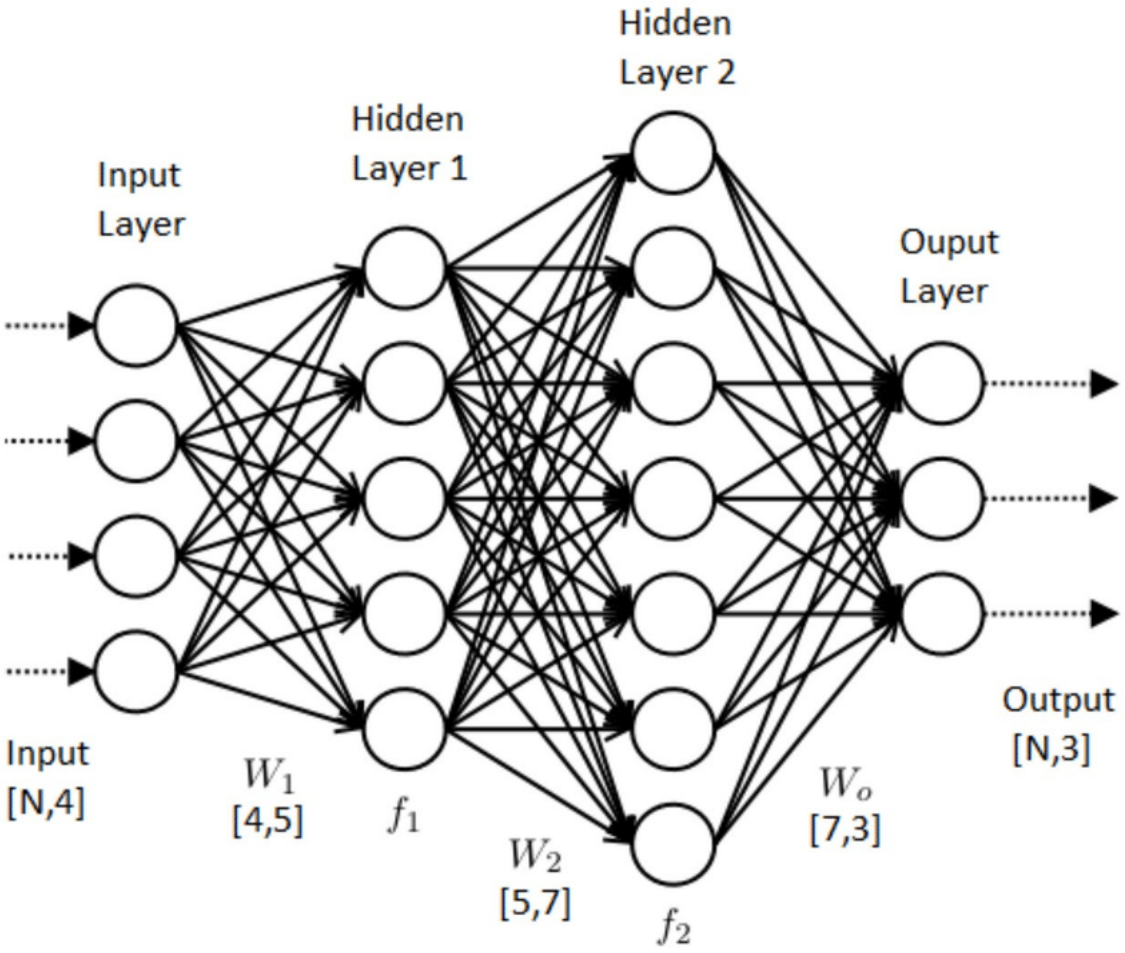
$$o = \tanh(w_۱x_۱ + w_۲x_۲ + \ldots + w_nx_n + b) \qquad (۴-۵)$$

[1] Bias

- تابع بیشینه هموار:

$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (۴-۶)$$

در ابتدا شبکه عصبی با وزن‌های تصادفی تعریف می‌شوند و سپس الگوریتم پس‌انتشار[1]، در هر دوره[2] از آموزش، وزن‌ها را بروزرسانی می‌کند. معماری شبکه عصبی در تصویر ۴-۵ مشخص است.



شکل ۴-۵: ساختار شبکه عصبی با ۲ لایه مخفی[۸]

[1] Backpropagation
[2] Epoch

۴-۱-۲ یادگیری بدون نظارت

در این روش یادگیری، داده‌ها بدون برچسب هستند و یا برچسب آن‌ها به مدل داده نمی‌شود. مدل خروجی‌ای پیش‌بینی نمی‌کند، بلکه الگوهای ساختاری داده را استخراج می‌کند. وظیفه این الگوریتم‌ها، پیدا کردن شباهت‌ها بین نمونه‌ها، خوشه‌بندی[۱] و یا پیدا کردن ناهنجاری‌ها[۲] در مجموعه داده است. از انواع این روش‌ها، می‌توان به تحلیل مولفه اساسی[۳]، خوشه‌بندی کی‌ـ میانگین[۴] و شبکه خودرمزگذار[۵] اشاره کرد.

۴-۲ معیارهای ارزیابی

عملکرد یک مدل جنبه مهمی است که نیاز به ارزیابی دقیق دارد. معیارهای ارزیابی، مقیاسی کمی از عملکرد یک شبکه عصبی یا مدل یادگیری عمیق در یک کار مشخص را ارائه می دهند. این معیارها به عنوان معیارهایی برای سنجش دقت، قابلیت اطمینان و اثربخشی عمل می‌کنند و به محققان و توسعه‌دهندگان کمک می‌کنند مدل‌ها را دقیق تنظیم کنند و تصمیمات آگاهانه بگیرند.

در زمینه یادگیری عمیق معیارهای ارزیابی گوناگونی طراحی شده که برای هدف‌ها و وظایف مختلف به کار می‌آید. در این بخش به معرفی آن‌ها می‌پردازیم.

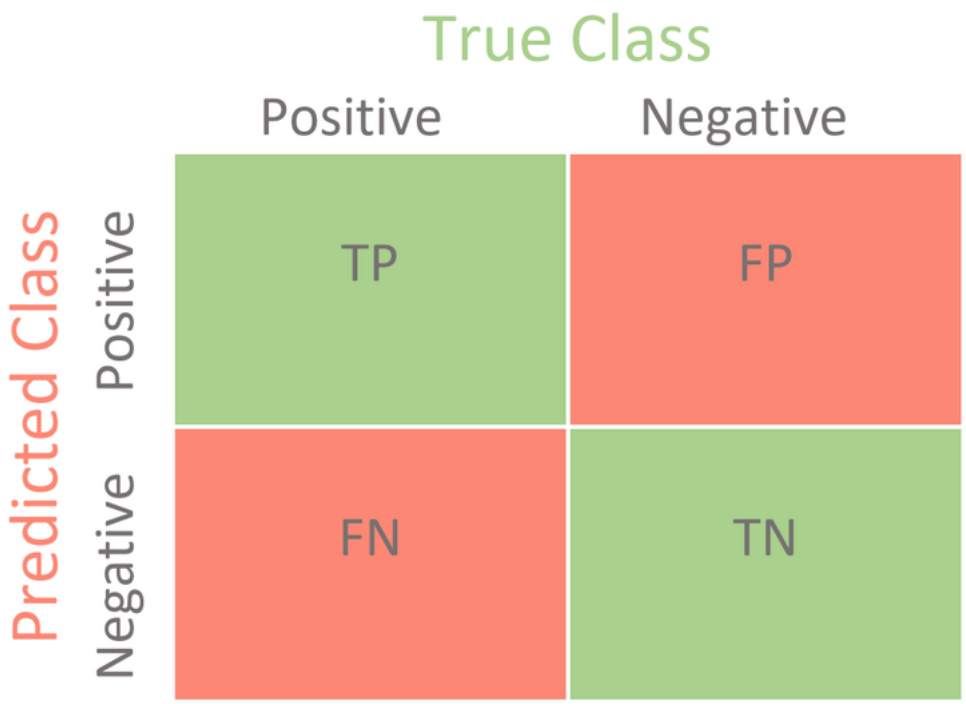
۱. ماتریس درهم‌ریختگی[۶]: ماتریسی می‌باشد که پیش‌بینی‌های مدل در مقایسه با برچسب واقعی داده‌ها نمایش می‌دهد. در مسائل دو کلاسه، ۴حالت زیر ممکن است که رخ بدهد:

- مثبت صادق(TP[7]): حالتی می‌باشد که کلاس داده مثبت و پیش‌بینی مدل نیز مثبت باشد.
- منفی صادق(TN[8]): حالتی می‌باشد که کلاس داده منفی و پیش‌بینی مدل نیز منفی باشد.
- مثبت کاذب(FP[9]): حالتی می‌باشد که کلاس داده منفی ولی پیش‌بینی مدل مثبت می‌باشد.

[1] Clustering
[2] Anomalies
[3] Principal Component Analysis
[4] K-means Clustering
[5] Autoencoder
[6] Confusion Matrix
[7] True Positive
[8] True Negative
[9] False Positive

- منفی کاذب(FN[1]): حالتی می‌باشد که کلاس داده مثبت ولی پیش‌بینی مدل منفی می‌باشد.

شکل ۴-۶ ساختار ماتریس درهم‌ریختگی نشان می‌دهد.



شکل ۴-۶: ساختار کلی ماتریس درهم‌ریختگی

۲. دقت[2]: به معنای نسبت تمام نمونه‌های درست پیش‌بینی شده به کل نمونه‌ها دقت می‌گویند.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (۴-۷)$$

۳. صحت[3]: نسبت تعداد نمونه‌های مثبتی که مدل پیش‌بینی کرده به کل نمونه‌هایی که مثبت پیش‌بینی

[1] False Negative
[2] Accuracy
[3] Precision

شده‌اند، صحت نام دارد.

$$Precision = \frac{TP}{TP + FP} \quad (۸-۴)$$

۴. بازیابی[1]: نسبت تعداد نمونه‌های مثبتی که مدل پیش‌بینی کرده به کل نمونه‌های مثبت در مجموعه داده، بازیابی نام دارد.

$$Recall = \frac{TP}{TP + FN} \quad (۹-۴)$$

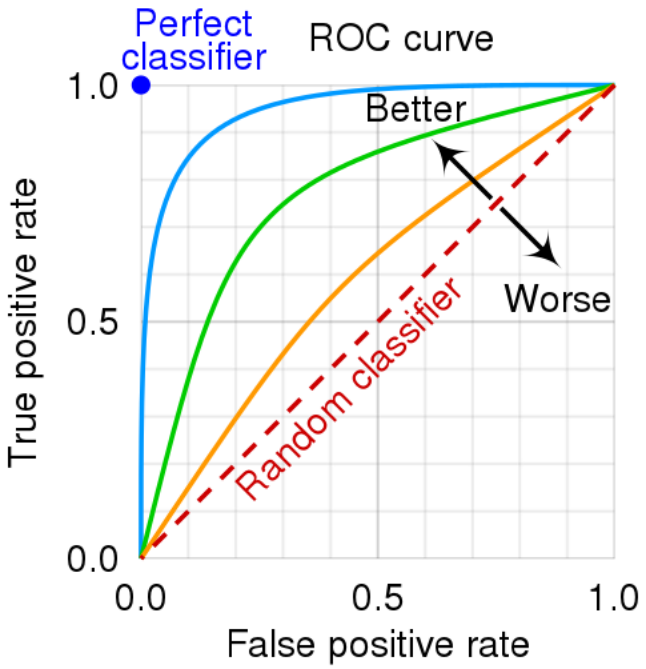
به این معیار نرخ مثبت صادق(TPR[2]) نیز گفته می‌شود. نرخ مثبت کاذب(FPR[3]) نیز تعریف می‌شود که نسبت $\frac{FP}{TN+FP}$ می‌باشد.

۵. امتیاز F1[4]: امتیاز F1، میانگین هم‌ساز[5] بین دو معیار صحت و بازیابی می‌باشد.

$$F۱ - Score = \frac{۲ \cdot Precision \cdot Recall}{Precision + Recall} \quad (۱۰-۴)$$

۶. سطح زیر منحنی عملیاتی گیرنده(AUC[6]): منحنی عملیاتی گیرنده[7] با رسم نرخ مثبت صادق در برابر نرخ مثبت کاذب در تنظیمات آستانه‌های مختلف ایجاد می‌شود. به سطح زیر این منحنی، AUC گفته می‌شود. هرچقدر مقدار AUC به ۱ نزدیک‌تر باشد، مدل عملکرد بهتری خواهد داشت. شکل ۴-۷چند مدل و منحنی ROC آن‌ها را مقایسه می‌کند.

[1] Recall
[2] True Positive Rate
[3] False Positive Rate
[4] F1-Score
[5] Harmonic Mean
[6] Area under the ROC Curve
[7] Receiver Operating Characteristic



شکل ۴-۷: مقایسه ROC چند مدل

۴-۳ شبکه مبدل

شبکه‌های مبدل در سال ۲۰۱۷ برای اولین بار در زمینه پردازش زبان‌های طبیعی و بهبود ترجمه ماشینی، معرفی شدند و انقلابی در حوزه یادگیری عمیق به وجود آوردند.[۳۷] مبدل‌ها با تکیه بر مکانیزم توجه به خود[۱] می‌توانند الگوها و وابستگی‌ها را در بین توالی داده‌ها بدست آورند. توانایی این مدل‌ها فقط به حوزه پردازش متن اختصاص نداشت و به حوزه‌های دیگر یادگیری عمیق، مانند پردازش تصویر وارد شد.[۱۴] در حوزه پردازش داده‌های جدولی، که مورد توجه ما هست نیز پژوهش‌هایی انجام شده است و شبکه مبدل توانایی خود را نشان داده است.[۱۸]

۴-۳-۱ مکانیزم توجه به خود

مکانیزم توجه به خود نوآوری شبکه‌های مبدل بودند که با استفاده از امتیاز توجه[۲] به قسمت‌های مختلف ورودی توجه بیشتر یا کمتری می‌کنند. در مسئله مورد نظر ما که داده‌ها جدولی می‌باشند، هر ردیف از جدول

[1] Self-attention
[2] Attention Score

نشان‌دهنده یک مشاهده یا نمونه می‌باشد که متشکل از ویژگی‌های پیوسته و طبقه‌بندی‌شده است. مکانیسم توجه به خود به مدل اجازه می‌دهد تا اهمیت هر ویژگی را با توجه به هر ویژگی دیگری در همان مشاهده بسنجد. برای هر ویژگی در یک ردیف معین، مکانیسم توجه به خود وزنی را برای هر ویژگی دیگر محاسبه می‌کند و تعیین می‌کند که مدل چقدر باید به هر ویژگی توجه کند. توجه به خود تعاملات بین ویژگی‌ها را با در نظر گرفتن روابط درون یک مشاهده ثبت می کند. این امر به ویژه در سناریوهایی که ترکیب‌های مشخصی از ویژگی‌ها برای انجام پیش‌بینی‌های دقیق بسیار مهم هستند، ارزشمند است و مدل‌های سنتی ممکن است برای گرفتن این وابستگی‌ها دچار مشکل شوند.[۱۸]

در معماری اولیه مبدل‌ها برای داده‌های جدولی که در [۱۸] معرفی شد، ابتدا ویژگی‌های طبقه‌بندی‌شده جاسازی[1] می‌شوند و به عنوان ورودی به چندین بلوک مبدل که روی هم قرار گرفته‌اند داده می‌شوند. خروجی مبدل تفسیری[2] است که از این ویژگی‌ها یاد می‌گیرد. در ادامه این تفسیر در کنار ویژگی‌های پیوسته که بهنجار شده‌اند به عنوان ورودی به یک شبکه عصبی چند لایه داده می‌شود و شبکه عصبی وظیفه دسته‌بندی را انجام می‌دهد. ساختار کلی معماری این در شکل ۴-۸ قابل مشاهده است.

در نسخه بهبود یافته این مدل که در [۱۷] معرفی شد، بر خلاف مدل ابتدایی که فقط ویژگی‌های طبقه‌بندی‌شده به مبدل داده می‌شود، این مدل هر دو نوع ویژگی‌ها یعنی پیوسته و طبقه‌بندی‌شده را به مبدل می‌دهد و بر اساس تفسیری که از آن‌ها بدست می‌آید دسته‌بندی می‌کند.از آنجایی که ممکن است ویژگی‌های پیوسته و طبقه‌بندی‌شده به هم مرتبط باشند این امر می‌تواند باعث بهبود عملکرد شود. در شکل ۴-۹ معماری این شبکه قابل مشاهده است. در شکل ۴-۱۰ نیز مقایسه‌ای از دو مدل TabTransfomer و FT-Transformer نشان داده شده است.

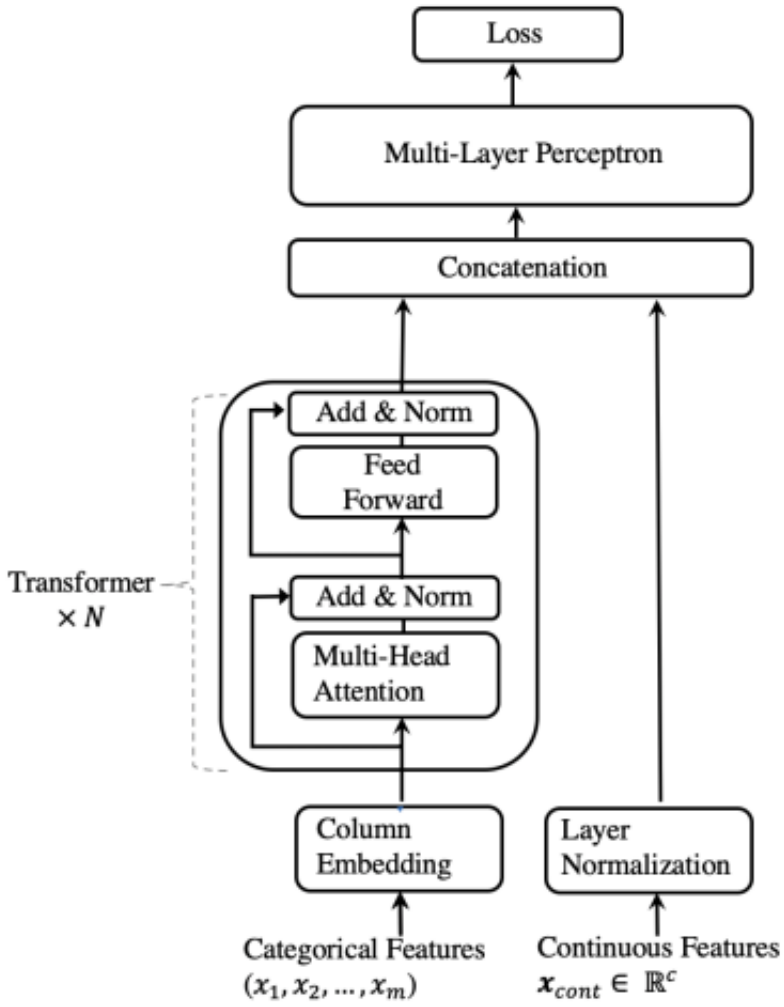
۴-۳-۲ پیاده‌سازی شبکه مبدل

برای پیاده‌سازی شبکه مبدل از کتاب‌خانه Pytorch-widedeep استفاده شده است که مستندات آن در این آدرس قابل مشاهده است. همانند بخش ۳-۲-۷ از ابزار Google Colab برای اجرا کدها استفاده شده است.

ابرپارامترهای این شبکه عبارتند از:

۱. تعداد بلوک‌های مبدلی برابر ۴ در نظر گرفته شده است.

[1] Embedding
[2] Representation



شکل ۴-۸: معماری مدل TabTransformer

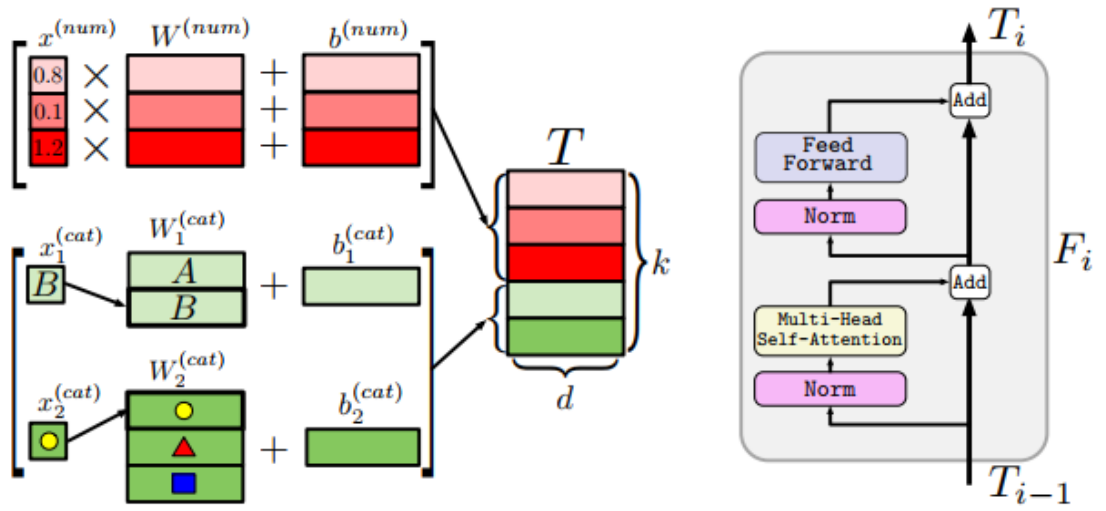
۲. برای جلوگیری از بیش‌برازش از روش حذف تصادفی با احتمال ۰.۲ استفاده شده است.

۳. اندازه بسته‌های ورودی ۲۵۶ در نظر گرفته شده است.

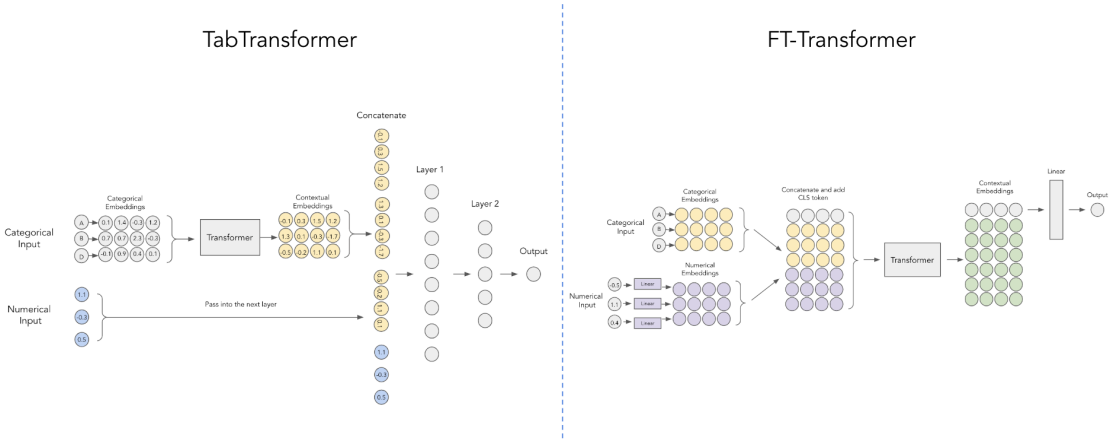
۴. تعداد دوره‌های آموزش برابر ۳۰ قرار داده شده است.

۵. از یک لایه مخفی در دسته‌بند استفاده شده است و تعداد نورون‌های لایه مخفی برابر ۱۰۲۴ در نظر گرفته شده است.

۶. از الگوریتم Adam و نرخ یادگیری $۱۰^{-۳}$ استفاده شده است.



شکل ۴-۹: معماری مدل FT-Transformer

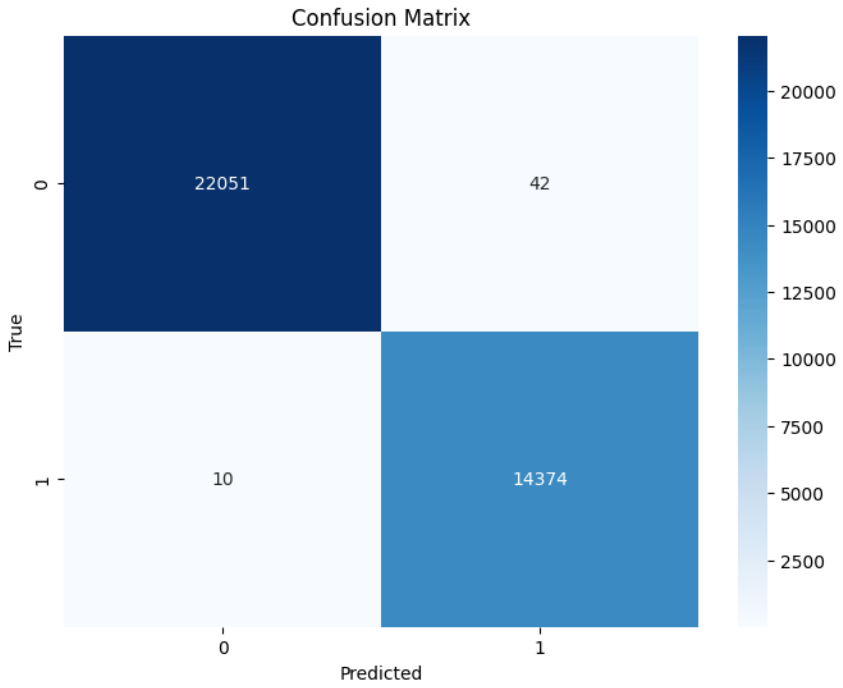


شکل ۴-۱۰: مقایسه TabTransfomer و FT-Transformer

داده‌های مصنوعی‌ای که در بخش ۳ تولید شدند با داده‌های اصلی ترکیب شدند و از مجموعه داده جدید و بهبودیافته برای آموزش مدل مبدل استفاده شد. در ادامه به نتایج بدست آمده از مدل می‌پردازیم.

۴-۳-۳ ارزیابی نتایج بدست آمده

مدل ابتدا بر روی داده‌ها آموزش یافت و بر روی مجموعه داده آزمایشی که در حین آموزش به آن دسترسی نداشت، ارزیابی شد.



شکل ۴-۱۱: ماتریس درهم‌ریختگی

ماتریس درهم‌ریختگی:

در شکل ۴-۱۱ ماتریس درهم‌ریختگی نمایش داده شده است.

دقت، صحت، بازیابی و امتیاز **F1**:

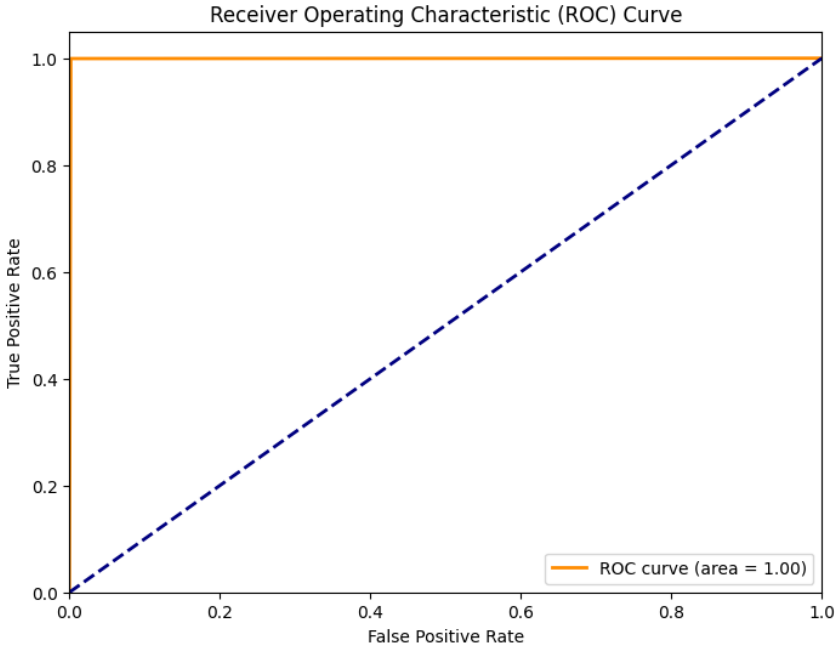
مدل به دقت 0.999، صحت 0.997، بازیابی 0.999 و امتیاز F1 0.998 دست یافت.

ROC و **AUC**:

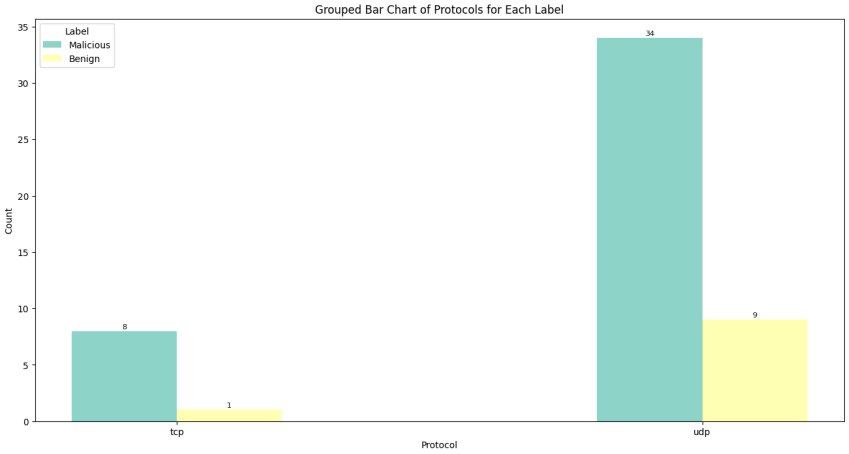
منحنی ROC و سطح زیر نمودار آن در شکل ۴-۱۲ نشان داده شده است.

اطلاعاتی درباره نمونه‌های اشتباه پیش‌بینی شده

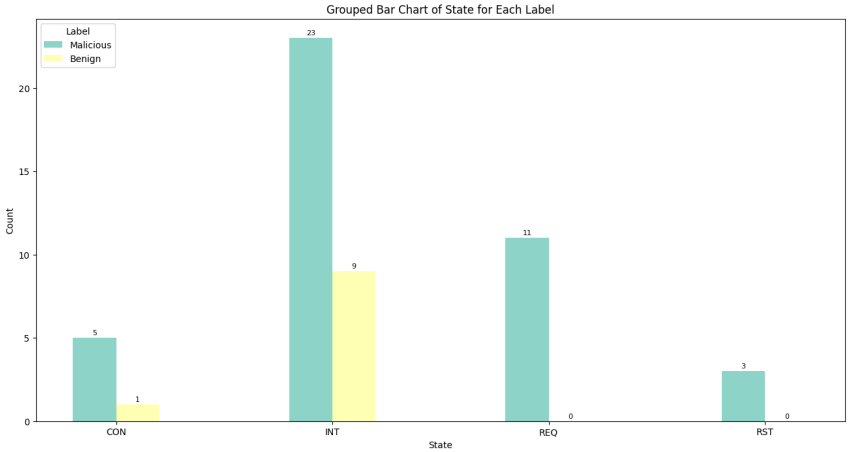
تعداد ۵۲ نمونه از مجموعه داده آزمایشی به اشتباه پیش‌بینی شده‌اند که در ادامه برخی از ویژگی‌های آن‌ها را در شکل‌های ۴-۱۳، ۴-۱۴ و ۴-۱۵ بررسی می‌کنیم.



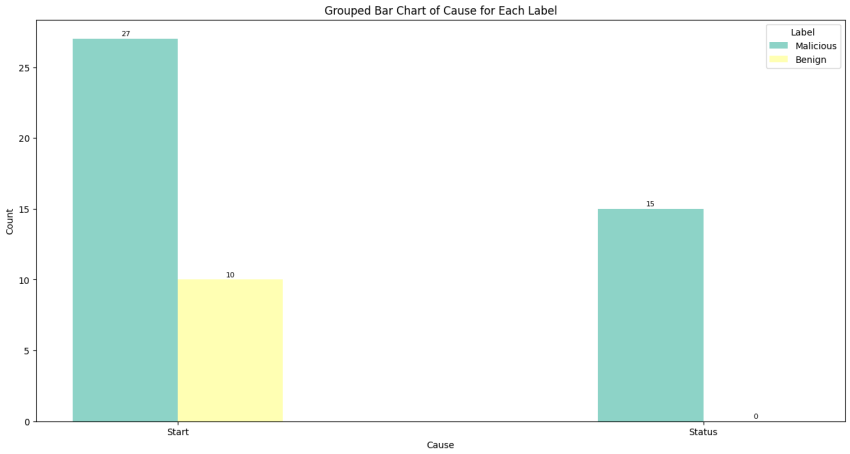
شکل ۴-۱۲: منحنی ROC



شکل ۴-۱۳: پروتکل نمونه‌هایی که اشتباه پیش‌بینی شده‌اند



شکل ۴-۱۴: State نمونه‌هایی که اشتباه پیش‌بینی شده‌اند



شکل ۴-۱۵: Cause نمونه‌هایی که اشتباه پیش‌بینی شده‌اند

فصل ۵

جمع‌بندی

در این پروژه عملکرد سامانه‌های تشخیص نفوذ با تمرکز بر ترافیک نسل ۵ در شبکه مورد بررسی قرار گرفت. سامانه‌های بسیاری برای تشخیص نفوذ توسعه یافته‌اند، اما با پیشرفت فناوری، حمله‌ها و نفوذها پیشرفت می‌کنند و تکامل می‌یابند. به همین منظور، سامانه‌های تشخیص نفوذ باید بروزرسانی شوند و مقاومت کافی در برابر تهدیدها را داشته باشند. یادگیری عمیق که حوزه‌ای همواره در حال پیشرفت است، به کمک توسعه‌دهندگان و محققان آمده که سامانه‌های تشخیص نفوذ بهتری آماده کنند. برای پیاده‌سازی الگوریتم‌های یادگیری عمیق، احتیاج به مجموعه داده‌ای داریم که هم از نظر کیفیت و هم از نظر کمیت قابل قبول باشد. بسیاری از مجموعه داده‌ها در حوزه تشخیص نفوذ از مشکل نامتوازنی رنج می‌برند و در این پروژه سعی بر این شد که با استفاده از شبکه‌های مولد متخاصم، راه‌حلی برای این مشکل ارائه شود. پس از برطرف کردن مشکل نامتوازنی، وارد بحث دسته‌بندی داده‌ها شدیم و با استفاده از مدل‌های مبتنی بر مبدل‌ها که از مدرن‌ترین و جدیدترین معماری‌های یادگیری عمیق هستند، داده‌های شبکه را به دو دسته حمله و بی‌خطر دسته‌بندی کردیم.

کتاب‌نامه

[1] Deep understanding of discriminative and generative models in machine learning. Accessed on: Sep. 30, 2023.

[2] Discriminative model. Accessed on: Sep. 30, 2023.

[3] Generative model. Accessed on: Sep. 30, 2023.

[4] KDD Cup 1999: Computer Network Intrusion Detection. Accessed: Sep. 15, 2023.

[5] Support vector machine. Accessed on: Oct. 3, 2023.

[6] Google developers - machine learning - gan generator, Publication Year 2022. Accessed on: Sep. 30, 2023.

[7] Abu Alfeilat, H. A., Hassanat, A. B., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., and Prasath, V. S. Effects of distance measure choice on k-nearest neighbor classifier performance: a review. *Big data 7*, 4 (2019), 221–248.

[8] Araujo, V., Guimarães, A., Campos Souza, P., Rezende, T., and Araujo, V. Using resistin, glucose, age and bmi and pruning fuzzy neural network for the construction of expert systems in the prediction of breast cancer. *Machine Learning and Knowledge Extraction 1* (02 2019).

[9] Belavagi, M. C., and Muniyal, B. Performance evaluation of supervised machine learning algorithms for intrusion detection. in *Proceedings of Computer Science* (January 2016), volume 89, pp. 117–123.

[10] Breiman, L. Random forests. *Machine learning 45* (2001), 5–32.

[11] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research 16* (2002), 321–357.

[12] Claise, B. Cisco systems netflow services export version 9. tech. rep., 2004.

[13] Costa, V. G., and Pedreira, C. E. Recent advances in decision trees: An updated survey. *Artificial Intelligence Review 56*, 5 (2023), 4765–4800.

[14] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[15] Ge, M., Syed, N. F., Fu, X., Baig, Z., and Robles-Kelly, A. Towards a deep learning-driven intrusion detection approach for internet of things. *Computer Networks 186* (February 2021), 107784.

[16] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks, 2014.

[17] Gorishniy, Y., Rubachev, I., Khrulkov, V., and Babenko, A. Revisiting deep learning models for tabular data, 2023.

[18] Huang, X., Khetan, A., Cvitkovic, M., and Karnin, Z. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678* (2020).

[19] Jabbar, A., Li, X., and Omar, B. A survey on generative adversarial networks: Variants, applications, and training, 2020.

[20] Jiang, Z., Pan, T., Zhang, C., and Yang, J. A new oversampling method based on the classification contribution degree. *Symmetry 13*, 2 (2021), 194.

[21] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization, 2017.

[22] Lewis, J. Economic impact of cybercrime, no slowing down. *Center for Strategic International Studies, McAfee 13* (2018), 2019.

[23] Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., and Dai, K. An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert Systems with Applications 39*, 1 (January 2012), 424–430.

[24] Manocchio, L. D., Layeghy, S., Lo, W. W., Kulatilleke, G. K., Sarhan, M., and Portmann, M. Flowtransformer: A transformer framework for flow-based network intrusion detection systems, 2023.

[25] Mirza, M., and Osindero, S. Conditional generative adversarial nets, 2014.

[26] Mohi-ud din, G. Nsl-kdd, 2018.

[27] NG, A. Train/dev/test sets - deep neural networks. Accessed on: Sep. 16, 2023.

[28] Palanivinayagam, A., and Damaševičius, R. Effective handling of missing values in datasets for classification using machine learning methods. *Information 14*, 2 (2023), 92.

[29] Piri, E., Ruuska, P., Kanstren, T., Mäkelä, J., Korva, J., Hekkala, A., Pouttu, A., Liinamaa, O., Latva-Aho, M., Vierimaa, K., et al. 5gtn: A test network for 5g application development and testing. in *2016 European Conference on Networks and Communications (EuCNC)* (2016), IEEE, p. 313–318.

[30] Potdar, K., Pardawala, T., and Pai, C. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications 175* (10 2017), 7–9.

[31] Qosient LLC. Argus: Network Flow Monitoring Tool. https://openargus.org/, v3.0.0. Accessed: Sep. 16, 2023.

[32] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans, 2016.

[33] Samarakoon, S., Siriwardhana, Y., Porambage, P., Liyanage, M., Chang, S.-Y., Kim, J., Kim, J., and Ylianttila, M. 5g-nidd: A comprehensive network intrusion detection dataset generated over 5g wireless network, 2022.

[34] Santoso, B., Wijayanto, H., Notodiputro, K., and Sartono, B. Synthetic over sampling methods for handling class imbalanced problems : A review. *IOP Conference Series: Earth and Environmental Science 58* (03 2017), 012031.

[35] Saputra, A., Hindarto, D., and Haryono, H. Supervised learning from data mining on process data loggers on micro-controllers. *Sinkron 8* (01 2023), 157–165.

[36] Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. in *4th International Conference on Information Systems Security and Privacy (ICISSP)* (Portugal, January 2018).

[37] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).

[38] Wikipedia contributors. Feature scaling — Wikipedia, the free encyclopedia, 2023. [Online; accessed 16-September-2023].

[39] Yekkehkhany, B., Safari, A., Homayouni, S., and Hasanlou, M. A comparison study of different kernel functions for svm-based classification of multi-temporal polarimetry sar data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 40* (2014), 281–285.

واژه‌نامه فارسی به انگلیسی

ابرپارامتر . . . . . Hyperparameter
ابرصفحه . . . . . Hyperplane
آزمایش . . . . . Test
از پیش آموخته . . . . . Pre-trained
امتیاز توجه . . . . . Attention Score
امنیت رایانه‌ای . . . . . Cybersecurity
آموزش . . . . . Train
انحراف . . . . . Bias
بازیابی . . . . . Recall
بستر آزمایشی . . . . . Testbed
بهنجار کردن . . . . . Normalize
بیش‌برازش . . . . . Overfitting
بیز ساده . . . . . Naive Bayes
پردازش تصویر . . . . . Image Processing
پردازش زبان‌های طبیعی . . . . . Natural Language Processing
پس‌انتشار . . . . . Backpropagation
پیش‌پردازش . . . . . Preprocessing
پیوسته . . . . . Continuous
تابع فعال‌سازی . . . . . Activation Function
تابع هدف . . . . . Objective Function
تابع هسته . . . . . Kernel Function
تحلیل مولفه اساسی . . . . . Principal Component Analysis
ترتیبی . . . . . Ordinal
تغییر مقیاس ویژگی‌ها . . . . . Feature Scaling
تفسیر . . . . . Representation
تفکیک‌کننده . . . . . Discriminator

واژه‌نامه فارسی به انگلیسی

توجه به خود ..... Self-attention
توزیع یکنواخت ..... Uniform Distribution
جاسازی ..... Embedding
جدولی ..... Tabular
جریان شبکه ..... Netflow
جنگل تصادفی ..... Random Forest
حذف تصادفی ..... Dropout
خودرمزگذار ..... Autoencoder
خودرمزگذار متغیر ..... Variational Autoencoder
خوشه‌بندی ..... Clustering
خوشه‌بندی کی‌ـ میانگین ..... K-means Clustering
درخت تصمیم ..... Decision Tree
درگاه ..... Port
درون‌یابی ..... Interpolating
دست‌دادن سه طرفه ..... Three Way Handshake
دسترس‌پذیری ..... Availability
دسته‌بند خطی ..... Linear Classifier
دقت ..... Accuracy
دنیای واقعی ..... Real-World
دوره ..... Epoch
رمزگذاری ..... Encoding
سامانه تشخیص نفوذ شبکه ..... Network Intrusion Detection System
سرآیند ..... Header
شبکه‌های مجازی شده ..... Virtualized Networks
شبکه مصنوعی ..... Neural Network
شبکه مولد متخاصم ..... Generative Adversarial Network
صحت ..... Precision
طبقه‌بندی‌شده ..... Categorical
غیراتصال‌گرا ..... Connection-less
گرادیان افزایشی ..... Gradient Ascent
گرادیان کاهشی ..... Gradient Descent
لایه بازگشتی ..... Recurrent Layer
لایه‌ی خطی ..... Linear Layer
لایه کاربرد ..... Application Layer

واژه‌نامه فارسی به انگلیسی

لایه همگشتی ..... Convolutional Layer
ماتریس درهم‌ریختگی ..... Confusion Matrix
ماشین بردار پشتیبان ..... Support Vector Machine
مبتنی بر بسته ..... Packet-Based
مبتنی بر جریان ..... Flow-Based
مثبت صادق ..... True Positive
مثبت کاذب ..... False Positive
مدل انتشاری ..... Diffusion Model
مدل‌های تمایزی ..... Discriminative Models
مدل‌های مولد ..... Generative Models
محرمانگی ..... Confidentiality
محروم‌سازی از سرویس ..... Denial of Service
منحنی عملیاتی گیرنده ..... Receiver Operating Characteristic
منفی صادق ..... True Negative
منفی کاذب ..... False Negative
مولد ..... Generator
میانگین هم‌ساز ..... Harmonic Mean
نامتوازن ..... Unbalanced
ناهنجاری ..... Anomaly
نرخ مثبت صادق ..... True Positive Rate
نرخ مثبت کاذب ..... False Positive Rate
نزدیک‌ترین همسایه ..... Nearest Neighbors
وایازش لجستیک ..... Logistic Regression
ویژگی ..... Feature
یادگیری بدون نظارت ..... Unsupervised Learning
یادگیری تجمیعی ..... Ensemble Learning
یادگیری عمیق ..... Deep Learning
یادگیری ماشین ..... Machine Learning
یادگیری نظارت‌شده ..... Supervised Learning
یکپارچگی ..... Integrity
یک‌داغ ..... One-hot

واژه‌نامه انگلیسی به فارسی

دقت ..... Accuracy
تابع فعال‌سازی ..... Activation Function
ناهنجاری ..... Anomaly
لایه کاربرد ..... Application Layer
امتیاز توجه ..... Attention Score
خودرمزگذار ..... Autoencoder
دسترس‌پذیری ..... Availability
پس‌انتشار ..... Backpropagation
انحراف ..... Bias
طبقه‌بندی‌شده ..... Categorical
خوشه‌بندی ..... Clustering
محرمانگی ..... Confidentiality
ماتریس درهم‌ریختگی ..... Confusion Matrix
غیراتصال‌گرا ..... Connection-less
پیوسته ..... Continuous
لایه همگشتی ..... Convolutional Layer
امنیت رایانه‌ای ..... Cybersecurity
درخت تصمیم ..... Decision Tree
یادگیری عمیق ..... Deep Learning
محروم‌سازی از سرویس ..... Denial of Service
مدل انتشاری ..... Diffusion Model
مدل‌های تمایزی ..... Discriminative Models
تفکیک‌کننده ..... Discriminator
حذف تصادفی ..... Dropout
رمزگذاری ..... Encoding
جاسازی ..... Embedding

واژه‌نامه انگلیسی به فارسی

یادگیری تجمیعی ..... Ensemble Learning
دوره ..... Epoch
منفی کاذب ..... False Negative
مثبت کاذب ..... False Positive
نرخ مثبت کاذب ..... False Positive Rate
ویژگی ..... Feature
تغییر مقیاس ویژگی‌ها ..... Feature Scaling
مبتنی بر جریان ..... Flow-Based
شبکه مولد متخاصم ..... Generative Adversarial Network
مدل‌های مولد ..... Generative Models
مولد ..... Generator
گرادیان افزایشی ..... Gradient Ascent
گرادیان کاهشی ..... Gradient Descent
میانگین هم‌ساز ..... Harmonic Mean
سرآیند ..... Header
ابرپارامتر ..... Hyperparameter
ابرصفحه ..... Hyperplane
پردازش تصویر ..... Image Processing
یکپارچگی ..... Integrity
درون‌یابی ..... Interpolating
خوشه‌بندی کی‌ـ میانگین ..... K-means Clustering
تابع هسته ..... Kernel Function
دسته‌بند خطی ..... Linear Classifier
لایه‌ی خطی ..... Linear Layer
وایازش لجستیک ..... Logistic Regression
یادگیری ماشین ..... Machine Learning
بیز ساده ..... Naive Bayes
پردازش زبان‌های طبیعی ..... Natural Language Processing
نزدیک‌ترین همسایه ..... Nearest Neighbors
شبکه مصنوعی ..... Neural Network
سامانه تشخیص نفوذ شبکه ..... Network Intrusion Detection System
جریان شبکه ..... Netflow
بهنجار کردن ..... Normalize
تابع هدف ..... Objective Function

واژه‌نامه انگلیسی به فارسی

یک‌داغ ..... One-hot
ترتیبی ..... Ordinal
بیش‌برازش ..... Overfitting
مبتنی بر بسته ..... Packet-Based
درگاه ..... Port
صحت ..... Precision
از پیش آموخته ..... Pre-trained
پیش‌پردازش ..... Preprocessing
تحلیل مولفه اساسی ..... Principal Component Analysis
جنگل تصادفی ..... Random Forest
دنیای واقعی ..... Real-World
لایه بازگشتی ..... Recurrent Layer
منحنی عملیاتی گیرنده ..... Receiver Operating Characteristic
بازیابی ..... Recall
تفسیر ..... Representation
توجه به خود ..... Self-attention
ماشین بردار پشتیبان ..... Support Vector Machine
یادگیری نظارت‌شده ..... Supervised Learning
جدولی ..... Tabular
آزمایش ..... Test
بستر آزمایشی ..... Testbed
دست‌دادن سه طرفه ..... Three Way Handshake
آموزش ..... Train
منفی صادق ..... True Negative
مثبت صادق ..... True Positive
نرخ مثبت صادق ..... True Positive Rate
توزیع یکنواخت ..... Uniform Distribution
نامتوازن ..... Unbalanced
یادگیری بدون نظارت ..... Unsupervised Learning
خودرمزگذار متغیر ..... Variational Autoencoder
شبکه‌های مجازی شده ..... Virtualized Networks

Abstract:

In today's world, cyber attacks have become widespread and can have catastrophic consequences for organizations and individuals. This is especially true in the era of 5G development, where the expansion and prevalence of these networks make securing them critically important. Destructive activities often exploit vulnerabilities in the network, making the development of resilient security measures essential. Intrusion detection systems have been developed as a defense mechanism against these threats, designed to monitor and identify unauthorized accesses, abnormal behaviors, and various types of attacks. However, the effectiveness of these systems is heavily dependent on the quality and quantity of the available training data. Many existing datasets for attack detection suffer from the unbalance problem, where certain types of attacks are underrepresented, leading to biased models and reduced system performance.

The goal of this project is to address the unbalanced nature of the datasets using deep learning methods, particularly focusing on generative adversarial networks for synthetic data generation and transformer models for classification. The aim is to achieve higher accuracy in the classification problem. To tackle the unbalance issue in the dataset, a model called the generative adversarial model is designed and trained on the dataset.

This model is trained to capture the patterns and features of both majority and minority classes and balance the dataset accordingly.

The synthetic generated data is then added to the original dataset, and by leveraging modern transformer-based architectures, the classification accuracy is improved on the new training data. This approach aims to enhance the performance of the intrusion detection system and make it more resistant to cyber attacks.

Keywords: Network Intrusion Detection System, 5G network, Deep Learning, Generative Adversarial Networks, Transformer, Unbalanced Dataset

Iran University of Science and Technology
Computer Engineering Department

Improving 5G Intrusion Detection with Synthetic Data Generation and Transformer-Based Networks

Bachelor of Science Thesis in Computer Engineering

By:

Sina Eskandari

Supervisor:

Dr. Vesal Hakami

October 2023