گزارش تحلیل دیتا ست NSL-KDD با یادگیری ماشین

**دیتا ست NSL-KDD یکی از معروف ترین مجموعه داده ها در تشخیص نفود شبکه (IDS- Intrusion Detection Systems) است.این دیتا شامل ویژگی های آماری از ارتباط شبکه و برچسب های مربوط به نوع ترافیک (نرمال یا انواع حمله) میباشد. حال با توجه به افزایش تبادل داده و گسترش اینترنت تهدید های مختلف مانند حملات R2L,Probe, Dos و U2R امنیت سیستم ها را در معرض خطر قرار میدهند . به همین دلیل استفاده از سیستم های تشخیص نفوذ (IDS- Intrusion Detection Systems)برای شناسایی و جلوگیری از حملات اهمیت زیادی دارد.**

**این دیتا ست یه نسخه بهبود یافتهKDD Cup 1999 است مشکلاتی مانند حجم بسیار زیاد و تکرار داده ها را ندارد.NSL KDD شامل 22543 اتصال شبکه هستش با تعدادی feature ها توصیف شده و برچسب(labels)یا اتصال عادی (normal)یا یکی از انواع حمله (attack)می باشد.**

**حالا هدف این پروژه چیست؟**

**هدف این پروژه این است مقایسه عملکرد دو الگوریتم Logistic Regression و Isolation Forest در تشخیص نفوذ میباشد**

**Isolation Forest (الگوریتم بدون نظارت برای شناسایی برای شناسایی ناهنجاری ها)**

**Logistic Regression(الگوریتم نظارت شده خطی)**

آماده سازی

**ابتدا داده NSL-KDD از سایت Kaggle دانلود و بارگزاری شد . مراحل پیش پردازش شامل موارد زیر بود:**

1. **بررسی تعداد رکورد ها و ستون های**
2. **حذف یا نادیده گرفتن مقادیر گمشده**
3. **تبدیل برخی Feature ها (به طور مثال protocol\_type,service,flag) به داده های عددی با one-Hot**
4. **تعریف جدید در Labels به normal و attack**
5. **تقسیم داده ها به دوبخش آموزش ئ تست (70% و 30%)**

اجرای الگوریتم

**1.isolation Forest**

**این الگوریتم یک روش unsupervised که به طور خاص برای تشخیص ناهنجاری طراحی شده .ایده آن این است که (مثل حملات ) راحت تر از داده های عادی جدا میشنوند . در کل اصل کار اینه که داده های normal (عادی) با داده های غیر عادی (مثل حملات سایبری ) رو از بقیه جدا کنی .**

**2.logistic regression**

**این الگوریتم یه مدل خطی برای طبقه بندی دودویی ست و یک روش supervised است . با استفاده از ویژگی های ورودی , یک مرض تصمیم بین کلاس ها ترسیم میکند . مدل روی داده های اموزشیfit شد و سپس روی داده ی تست predict اجرا گردید . برای جلوگیری از مشکل همگرایی پارامتر max\_iter = 10000 در نظر گرفته شده (پ.ن: از اونجایی که به طور دیفالت 1000 هستسش مدل موقع بهینه سازی جواب خوبی نمیداد. پس 10000 و 5000 گزینه مناسب تری است)**

ارزیابی مدل

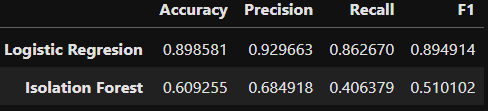
**برای ارزیابی مدل ها از معیار های زیر استفاده شده است:**

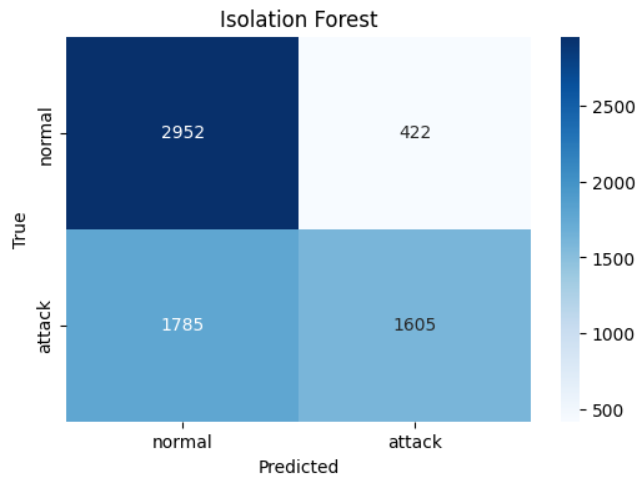
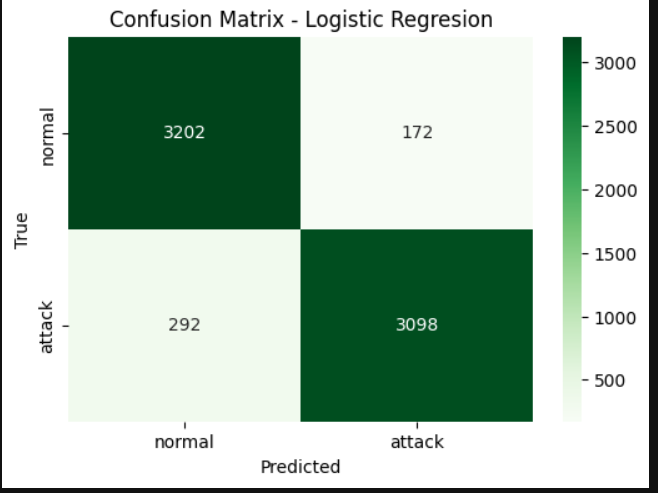
**1.Accuracy: درصد پیش بینی صحیح کل نمودار ها**

**2.Precision: دقت شناسایی حملات**

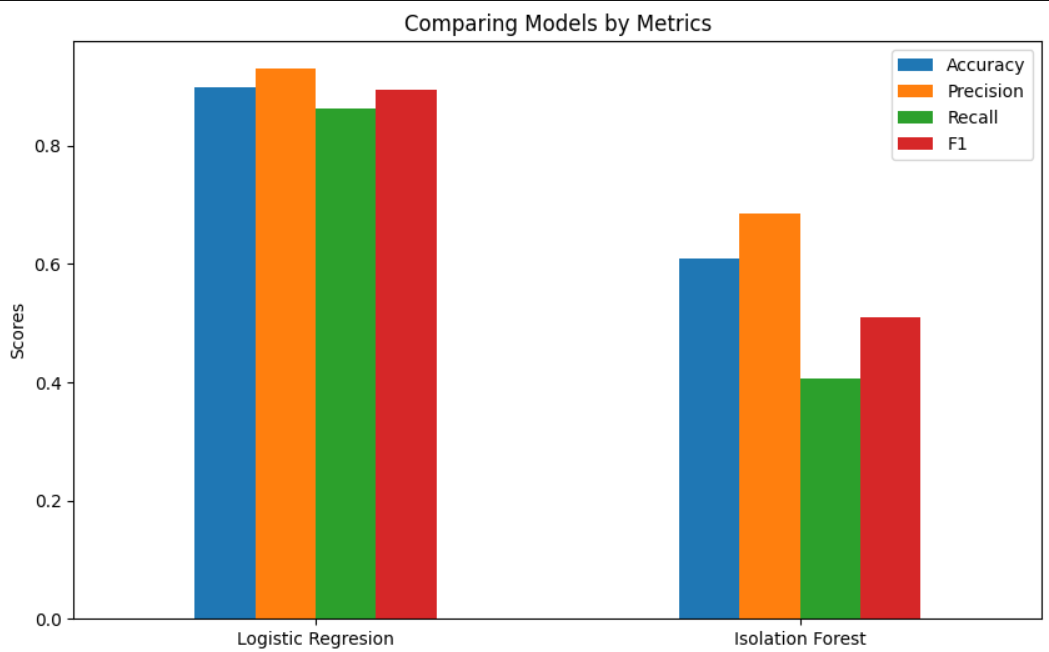
**3.Recall: توانای در پیدا کردن تمام حملات**

**4.F1-Score: میانگین هماهنگ Precision , Recall**

**نتایج مقایسه :**

**یه bar plot هم رسم شد که نشان داد Logistic regression در تمام معیار ها عمکرد بهتری نسبت به Isolation Forest دارد.همچنین یه Heatmap برای دقت الگوریتم و اختلاف نتایج آن ها در صفحه بعد نمایش داده میشود.**

**Bar plot**



نتیجه گیری

**در این پروژه NSL-KDD ست مورد بررسی قرار گرفت و دو الگوریتم Logistic Regression و Isolation Forest بر روی ان اجرا شد و هدف این بود که با استفاده از دو روش Supervised و Unsupervised بررسی کرد که کدام روش و با کدام الگوریتم دقت بالاتر و کیفیت و نتایج بهتری در اختیار ما قرار میدهد. بررسی های نشان میدهد :**

**Logistic Regression: با توجه استفاده از برچسب های واقعی توانست دقت بالاتری در شناسایی حملات بدست اورد**

**Isolation Forest: به دلیل Unsupervised بودن عملکرد ضعیف تری داشت اما همچنان توانست الگوهای ناهنجاری را تشخیص دهد**

**حال برای بهبود عملکرد میتوان از الگوریتم های SVM و Random Forest یا شبکه های عصبی های مصنوعی عمیق استفاده کرد.(در پیوست های مقاله های که این الگوریتم هارا استفاده کرده اند به اشتراک گذاشته میشوند.**

**Linear Regression:** <https://scikit-learn.org/1.5/auto_examples/linear_model/plot_ols.html>

**Isolation Forest:** <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html>

**Label Encoding:** <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>

**ScikitLearn:** <https://scikit-learn.org/stable>

**Seaborn plots:** <https://seaborn.pydata.org/examples/index.html>

**Scores:** <https://scikit-learn.org/stable/modules/model_evaluation.html>

**Pro:** <https://www.geeksforgeeks.org/machine-learning/ml-models-score-and-error>

**مقاله ها(chat\_GPT5) :**

**Intrusion Detection algorithm based on improved SVM Classification method:**[**https://www.researchgate.net/publication/359796437\_An\_Analysis\_of\_Intrusion\_Detection\_Classification\_using\_Supervised\_Machine\_Learning\_Algorithms\_on\_NSL-KDD\_Dataset?utm\_source=chatgpt.com**](https://www.researchgate.net/publication/359796437_An_Analysis_of_Intrusion_Detection_Classification_using_Supervised_Machine_Learning_Algorithms_on_NSL-KDD_Dataset?utm_source=chatgpt.com)

**Intelligent intrusion Detection system using Rf, SVM and DT:**[**https://thescipub.com/abstract/jcssp.2025.1749.1759?utm\_source=chatgpt.com**](https://thescipub.com/abstract/jcssp.2025.1749.1759?utm_source=chatgpt.com)

**Intrusion Detection System using Support vector Machine (SVM)on the KDDCUP99 and NSL-KDD datasets:**[**https://arxiv.org/abs/2209.05579?utm\_source=chatgpt.c**](https://arxiv.org/abs/2209.05579?utm_source=chatgpt.c)

**Hatef jani**

**Email:** [janalipourhatef@gmail.com](mailto:janalipourhatef@gmail.com)

**Github:Hatef-skywalker007**

**Thanks For Everything mispython**