به نام خدا

پروپوزال شرح عملکرد مسابقه AI

معرفی اعضای تیم

علی کهن، مهندس هوش مصنوعی

بنده مهندس هوش مصنوعی با بیش از ۶ سال تجربه در برنامه‌نویسی و بیش از ۴ سال سابقه تخصصی در حوزه هوش مصنوعی هستم. در طول این مدت، هم در صنعت و هم در عرصه آکادمیک پروژه‌های مختلفی را به انجام رسانده‌ام. همچنین در شرکت‌های داخلی و بین‌المللی به عنوان مهندس هوش مصنوعی و مهندس نرم‌افزار تجربه کسب کرده‌ام. حوزه‌های تخصصی من شامل یادگیری ماشین، یادگیری عمیق، بینایی ماشین، پردازش تصویر، مدل‌های زبانی بزرگ (LLMs) و الگوریتم‌های تکاملی می‌باشد. همچنین، مقالات متعددی در زمینه هوش مصنوعی منتشر کرده‌ام و به عنوان داور در ژورنال معتبر (IEEE access Q1) نیز فعالیت داشته‌ام.

محسن سالاری کردی، مهندس نرم افزار و فعال حوزه هوش مصنوعی

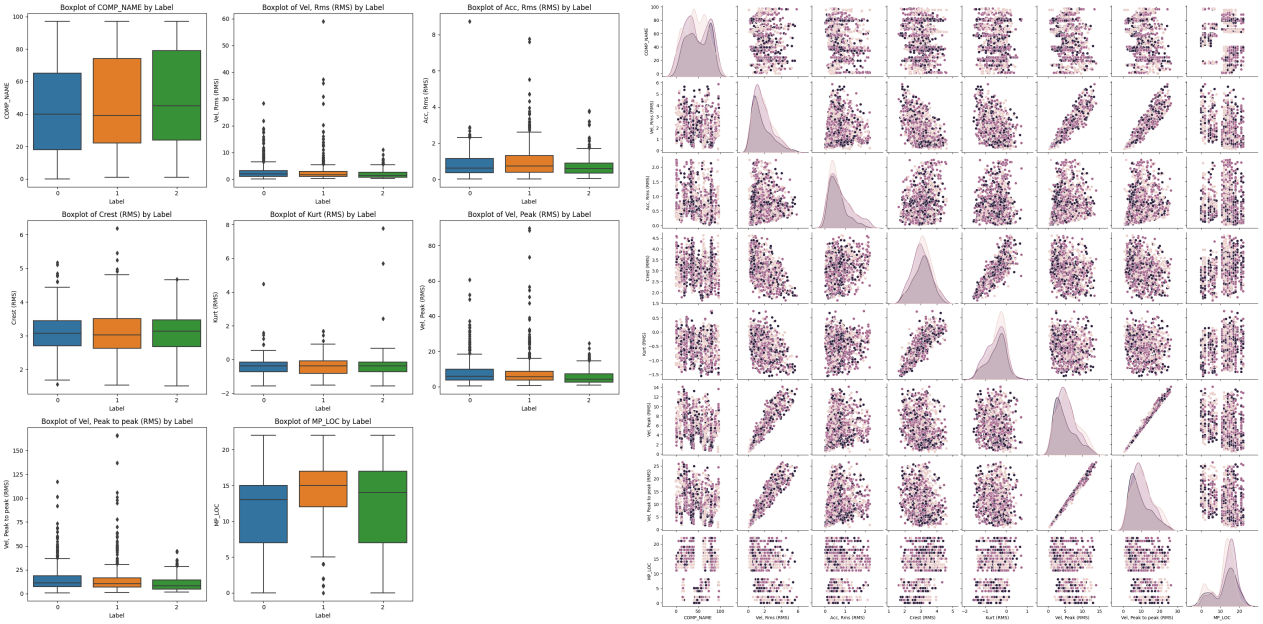
مهندس نرم‌افزار با بیش از ۶ سال تجربه در توسعه نرم‌افزار و بیش از ۴ سال سابقه تخصصی در زمینه‌های مختلف هوش مصنوعی. از دوران دانشگاه به هوش مصنوعی علاقه‌مند بوده و در پروژه‌های متعدد فعالیت داشته‌ام. در زمینه‌هایی نظیر یادگیری ماشین، یادگیری عمیق، پردازش زبان طبیعی، و مدل‌های زبانی بزرگ (LLMs) مهارت دارم. همچنین به‌عنوان یک توسعه‌دهنده سینیور بک‌اند، توانمندی‌هایی در حوزه DevOps و زیرساخت‌های ابری (Amazon Cloud و Google Cloud) کسب کرده‌ام. یکی از افتخارات من، حضور به‌عنوان نماینده ایران در Global GITEX 2023 به عنوان عضوی از پاویون ایرانیان نمایشگاه و همکاری به عنوان مدیر فنی بخش سرور با تیم‌های بین‌المللی در کشور های حاشیه خلیج فارس است.

بررسی اولیه دیتاست

در بررسی اولیه داده‌ها، مشخص شد که دیتاست ارائه‌شده از نوع ساختاری (structural) است و نیازمند انتخاب مدلی متناسب با این نوع داده‌هاست. برای تحلیل عمیق‌تر، تحقیقاتی روی پارامترهای مرتبط با خرابی بلبرینگ (bearing failure) انجام شد که نشان می‌دهد شاخص‌های "کرست فکتور" و "کورتوسیس" نقش کلیدی در تشخیص سلامت ماشین‌آلات دارند.

کرست فکتور نسبت پیک سیگنال به میانگین آن را اندازه می‌گیرد و افزایش این مقدار معمولاً نشان‌دهنده آسیب یا ضربات غیرعادی است. **کورتوسیس** به توزیع نوسانات کوچک پیرامون یک پیک اصلی اشاره دارد؛ مقادیر غیرعادی این شاخص معمولاً به خستگی یا نقص در دستگاه مربوط می‌شوند. این پارامترها با پایش شرایط عملیاتی دستگاه، امکان شناسایی زودهنگام مشکلات و برنامه‌ریزی برای اقدامات پیشگیرانه را فراهم می‌کنند.

با رسم scatter plot داده‌ها، متوجه شدیم که داده‌ها به‌شدت درهم‌تنیده بوده و کلاسه‌بندی آنها احتمالاً با روش‌های ساده یا مدل‌های کلاسیک، به‌سادگی امکان‌پذیر نیست. با استفاده از نمودار جعبه‌ای (Boxplot)، تعدادی داده‌ی پرت (Outlier) شناسایی شدند. هرچند این داده‌ها ممکن است واقعا پرت نباشند، به همین دلیل مدل را هم با حذف این داده‌ها و هم بدون حذف آن‌ها آزمایش کردیم.



(جهت مشاهده تصویر در مقیاس بزرگتر به notebook مراجع شود)

فاز تحقیقات (Research Phase)

پس از بررسی دیتاست، مدتی را به فاز تحقیق و جستجوی مقالاتی که در حوزه مشابه کار کرده‌اند اختصاص دادیم. در مقاله‌ی [1] تحقیقی توسط دانشگاه شریف انجام شده که در آن با استفاده از یک مدل CNN با بهبود 16 درصدی نسبت به روش deep FFNN به دقت 98 رسیده‌اند. آن ها با استفاده از سیگنال خام و بدون هیچگونه استخراج ویژگی به این دقت رسیده بودند که ما به علت عدم دسترسی به داده های خام نتوانستیم از این روش استفاده کنیم.

در مقاله‌ی دیگری [2] از روش deep belief network استفاده شده بود. ویژگی‌هایی که در این مقاله استخراج شده بود با ویژگی های دیتاست مقداری متفاوت بود؛ با این وجود ما تلاش کردیم که مدل را روی دیتاست اجرا کنیم اما نتایج مطلوبی حاصل نشد.

جهت مشاهده رفرنس‌ها به پایان گزارش مراجعه کنید.

انتخاب ویژگی‌ها (Feature Selection)

روش‌های متعددی برای انتخاب ویژگی وجود دارند که نسبت به حجم و نوع داده می‌توان روش مناسب را انتخاب کرد. با توجه به اندازه‌ی نسبتا کوچک دیتاست، ما از الگوریتم جستجوی کامل (exhaustive search) استفاده کردیم. این روش با جستجو در بین تمام احتمالات ممکن برای انتخاب ویژگی‌ها، به ما این امکان را می‌دهد که به بهینه سراسری (global optimum) دست یابیم و بهترین نتیجه ممکن را کسب کنیم. همچنین، متد تحلیل مولفه‌های اصلی (PCA) برای کاهش ابعاد داده‌ها نیز آزمایش شد، اما نتایج حاصل از آن رضایت‌بخش نبود و به همین دلیل در پایپلاین نهایی از این روش صرف‌نظر گردید.

مدل‌سازی (Modeling Phase)

در فاز مدلسازی این پروژه، مدل‌های مختلفی مانند درخت تصمیم، شبکه عصبی ،XGBoost ،KNN ،SVM Random Forest ،LightGBM مورد آزمایش قرار گرفتند. برای بهبود عملکرد این مدل‌ها، ابتدا نرمال‌سازی داده‌ها برای مدل‌هایی مانند شبکه عصبی که به مقیاس‌گذاری حساس هستند، در بازه ۰ تا ۱ انجام گرفت. برای سایر مدل‌ها، در صورت نیاز از دیگر تکنیک‌های پیش‌پردازش داده استفاده شد. همچنین برای بهینه‌سازی عملکرد، بهینه‌سازی هایپرپارامترها برای بعضی مدل‌ها انجام شد. برای این کار از روش grid search استفاده شد.

نتایج حاصل از تست مدل‌ها در جدول زیر ارائه شده است. پس از تجزیه و تحلیل دقیق نتایج، مشخص شد که مدل XGBoost بهترین عملکرد را از نظر دقت و کارایی نسبت به سایر مدل‌ها نشان می‌دهد که پس از اعمال بهینه‌سازی‌های لازم، به‌عنوان مدل نهایی انتخاب گردید.

Model Name	F1 Score
XGBoost	72

Model Name	F1 Score
KNN	63
SVM	64
Random Forest	67
LightGBM	59
Neural Network	64

گزارش نتایج (Results Report)

با استفاده از الگوریتم انتخاب شده XGBoost، مدل به میزان نسبتا خوبی قادر به پیش‌بینی کلاس‌ها بوده است. نتایج حاصله از confusion matrix در جدول زیر نمایش داده شده است.

Label	F1-Score	TP	FP	FN
Class 0	0.78	86	28	21
Class 1	0.75	81	30	23
Class 2	0.56	33	19	33

طراحی پایپ‌لاین (Pipeline Design)

پایپ‌لاینی که برای توسعه و آموزش مدل استفاده شده است، شامل سه مرحله به شرح زیر می‌باشد:

1. **پاک‌سازی داده‌ها (Data Cleaning):** در این مرحله، داده‌های پرت حذف می‌شوند. همچنین در صورت نیاز عمل نرمال‌سازی انجام می‌شود.

2. **انتخاب ویژگی‌ها (Feature Selection):** در این مرحله، ویژگی‌های کلیدی و موثر که بیشترین تاثیر را در عملکرد مدل دارند، شناسایی و انتخاب می‌شوند. این فرایند به منظور کاهش پیچیدگی مدل، بهبود سرعت پردازش و افزایش دقت مدل انجام می‌شود.

3. **آموزش مدل (Model Training):** در نهایت، در این مرحله مدل XGBoost را می‌توان با استفاده از داده‌های تمیز و ویژگی‌های انتخاب‌شده آموزش داد. هر چند در تست‌های گرفته شده، پاکسازی داده‌ها تاثیر چندانی در دقت مدل نداشت.

اپلیکیشن

جهت تسهیل فرایند تست مدل برای تیم داوری، یک اپلیکیشن با رابط کاربری گرافیکی (GUI) پیاده‌سازی شده است که امکان ارزیابی مدل را در **دو حالت** مختلف فراهم می‌آورد. در حالت اول، کاربران قادر به تست مدل با ورود دستی پارامترها هستند. در حالت دوم، کاربر می‌تواند داده‌ها را به صورت فایل CSV با فرمت مشابه دیتاست بارگذاری کنند. ***در صورتی که فایل CSV شامل برچسب‌ها (labels) باشد***، نرم‌افزار قبل از ارسال داده‌ها به مدل، برچسب‌ها را حذف کرده و پس از اجرای تست، خروجی مدل را با برچسب‌ها مقایسه می‌کند تا **درصد موفقیت مدل در تشخیص به‌طور دقیق محاسبه گردد.** سطرهای خطا دار با پس‌زمینه قرمز رنگ مشخص می‌شوند. در صورت عدم وجود برچسب در فایل CSV، نرم‌افزار خروجی مدل را برای هر سطر نمایش می‌دهد. این اپلیکیشن امکان تست سریع و دقیق مدل را با رابط کاربری ساده و کاربرپسند فراهم می‌کند.

مشارکت

علی کهن: فاز تحقیقات آکادمیک، تحلیل داده‌ها، انتخاب و بهینه‌سازی مدل

محسن سالاری: پرس‌وجو از متخصصین فنی حوزه، توسعه برنامه اجرایی و UI، تحلیل اولیه داده‌ها

خلاقیت و نوآوری - پیشنهادات آینده

در چالش فعلی، داده‌های ما از قبل استخراج ویژگی شده بودند که این مسئله باعث ایجاد محدودیت در خلاقیت و نوآوری ما شده بود. برای نمونه استفاده از مدل CNN روی داده‌های خام، ایجاد نگاه sequential و اعمال مدل‌های سری زمانی (مثل RNN, LSTM و حتی Transformer) می‌تواند دقت مدل را بهبود قابل توجهی بدهد.

ساختار فایل‌های ارسالی

فایل‌های ارسالی پروژه شامل موارد زیر هستند:

- **یک فایل Jupyter Notebook** که شامل تحلیل داده‌ها و فرآیندهای مربوط به تجسم داده‌ها (Data Visualization)، به اضافه آموزش مدل است.
- **یک فایل GUI** که نرم‌افزار تست مدل را شامل می‌شود.
- **یک فایل requirements.txt** که پکیج‌های مورد نیاز برای اجرای صحیح پروژه را فهرست می‌کند.
- **فایل پروپوزال PDF** (فایل فعلی) که توضیحات و اهداف پروژه را ارائه می‌دهد.
- **فایل توضیحات ویدیویی** که فرآیند و نحوه استفاده از اپلیکیشن را شرح می‌دهد.

منابع بخش تحقیقات

[1] Behzad, M., Izanlo, H., Davoodabadi, A., & Arghand, H. A. (2021). Fault detection of rolling element bearing using a temporal signal with artificial intelligence techniques. *Journal of Theoretical and Applied Vibration and Acoustics*, *7*(1), 55-71.

[2]Shao, H., Jiang, H., Zhang, X., & Niu, M. (2015). Rolling bearing fault diagnosis using an optimization deep belief network. *Measurement Science and Technology*, *26*(11), 115002.