



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

عنوان:

بهبود قابلیت اطمینان سامانه های ذخیره سازی داده

اعضای گروه

امیر ارسلان یاوری
سهند ذوفن

نام درس

سامانه های پیشرفته ذخیره سازی داده

نیم سال اول ۱۴۰۳-۱۴۰۴

نام استاد درس

دکتر حسین اسدی



چکیده

این پروژه به بهبود قابلیت اطمینان سامانه‌های ذخیره‌سازی داده با تحلیل داده‌های بدست آمده از ابزار SMART می‌پردازد. ابتدا مفاهیم اولیه مربوط به مکانیسم‌های خرابی دیسک‌های حالت جامد (SSD) تعریف می‌شود. سپس مجموعه داده‌های موجود مورد بررسی قرار گرفته و با استفاده از مدل‌های هوش مصنوعی، روابط موجود در داده‌ها تحلیل می‌شود. در پایان، نتایج بدست آمده تحلیل و به بررسی کاربردهای آن‌ها در پیش‌بینی نگهداری و افزایش قابلیت اطمینان سامانه‌های ذخیره‌سازی پرداخته می‌شود.

کلیدواژه‌ها: دیسک‌های حالت جامد SSD، ابزار SMART، هوش مصنوعی، زمان خرابی، مقدار آستانه، ناهنجاری، همبستگی میان ویژگی‌ها

۱-۰ مقدمه

۱-۱-۰ تعریف مسئله

دیسک‌های حالت جامد (SSD) به دلیل ویژگی‌های خاص ساختاری خود، در طول زمان ممکن است دچار خرابی‌ها و مشکلات مختلفی شوند. این خرابی‌ها می‌تواند ناشی از عواملی نظیر نشت داده‌ها، تعداد دفعات خواندن و نوشتن بیش از حد، یا حتی عوامل محیطی مانند دما و رطوبت باشد. این مشکلات به‌ویژه در سامانه‌های ذخیره‌سازی داده‌های حساس می‌تواند تأثیرات جدی بر قابلیت اطمینان و عملکرد سیستم‌ها بگذارد. ابزار SMART (Self-Monitoring, Analysis and Report- ing Technology) که در کنترل‌کننده دیسک‌ها تعبیه شده، به‌عنوان یک ابزار حیاتی برای نظارت بر وضعیت سلامت دیسک‌ها و پیش‌بینی خرابی‌های احتمالی، اطلاعات ارزشمندی را از قبیل حجم داده‌های خوانده‌شده و نوشته‌شده، مدت زمان کارکرد دیسک و تعداد خطاها فراهم می‌آورد. هدف این پروژه تحلیل و بهبود قابلیت اطمینان سامانه‌های ذخیره‌سازی داده است که بر اساس اطلاعات حاصل از ابزار SMART کار می‌کنند. در این راستا، با استفاده از یک مجموعه داده متشکل از ۵ هزار دیسک، شامل هزار دیسک خراب و ۴ هزار دیسک سالم، تلاش می‌شود تا روش‌هایی برای پیش‌بینی زمان خرابی دیسک‌های سالم، تعیین مقادیر آستانه برای ویژگی‌های مختلف SMART و تشخیص رفتارهای ناهنجار دیسک‌ها طراحی و پیاده‌سازی شود. در نهایت، همبستگی میان ویژگی‌های مختلف SMART برای درک بهتر علل خرابی‌ها و پیش‌بینی وقوع آن‌ها بررسی خواهد شد. این پروژه با استفاده از مدل‌های تحلیلی و هوش مصنوعی به استخراج الگوها و روابط موجود در داده‌ها پرداخته و راهکارهایی برای بهبود عملکرد و قابلیت اطمینان سامانه‌های ذخیره‌سازی داده ارائه خواهد کرد.

۲-۱-۰ ساختار گزارش

مقدمه: تشریح زمینه و اهمیت تحقیق، هدف از کشف روابط خرابی دیسک‌های حالت جامد. مفاهیم اولیه: در این قسمت به معرفی کوتاه مباحثی که در ادامه به آنها نیاز داریم خواهیم پرداخت و مفاهیم پایه‌ای مرتبط با گزارش را عنوان خواهیم کرد. پیاده‌سازی: در این قسمت به شرح پیاده‌سازی‌های صورت گرفته در جهت حل مسئله خواهیم پرداخت.

نتیجه‌گیری: جمع‌بندی نتایج حاصل از یادگیری مدل‌ها و تحلیل روابط و توسعه‌های انجام شده و ارائه دیدگاه‌ها و پیشنهادات برای تحقیقات آتی در قسمت پایانی عنوان شده‌اند.

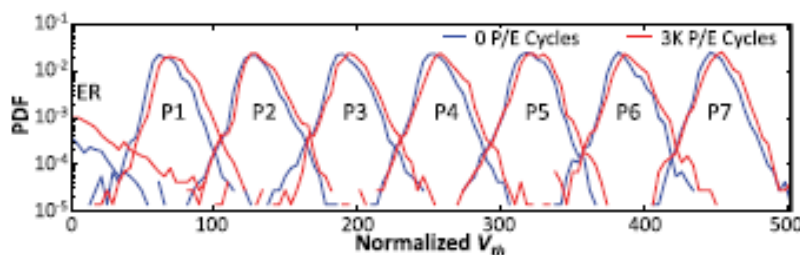
۲-۰ مفاهیم اولیه

در این بخش، ابتدا به معرفی و توضیح مفاهیم اساسی مرتبط با پروژه خواهیم پرداخت. همچنین نتایج و اطلاعات به دست آمده از تحقیقات پیشین را بررسی کرده و نحوه ارتباط آن‌ها با پروژه خود را تحلیل خواهیم کرد.

۱-۲-۰ خطا و مشکلات در درایوهای حالت جامد (SSD) و روش تشخیص آن‌ها

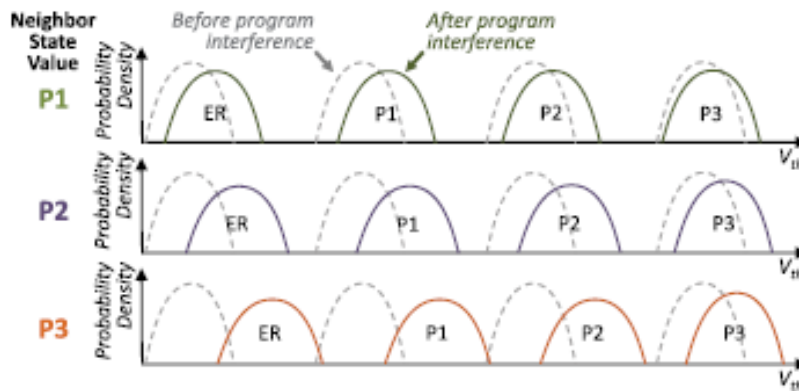
انواع خطاها و مشکلات در درایو حالت جامد

یکی از مهم‌ترین مشکلات در SSDها، خطاهای سیکل برنامه‌ریزی/پاک کردن است. هر سلول حافظه در طول عمر خود می‌تواند تعداد محدودی عملیات نوشتن و پاک کردن را تحمل کند. در هر سیکل P/E، لایه اکسید تونلی سلول دچار فرسایش می‌شود و الکترون‌ها در آن به دام می‌افتند. این فرآیند باعث تغییر تدریجی در توزیع ولتاژ آستانه سلول می‌شود و در نهایت ممکن است توانایی سلول در نگهداری صحیح داده‌ها از بین برود. این مشکل در فناوری‌های جدید که از سلول‌های کوچکتر استفاده می‌کنند، حادث‌تر است؛ چرا که تعداد الکترون‌های ذخیره‌شده در هر سلول به حدود ۱۰۰ الکترون کاهش می‌یابد.



شکل ۱: مقایسه دو حالت چرخه نوشتن/پاک کردن ۰ و ۳۰۰۰ بار

تداخل برنامه‌نویسی و تداخل سلول به سلول، دو مشکل مرتبط اما متمایز هستند که بر عملکرد SSD تأثیر می‌گذارند. تداخل برنامه‌نویسی زمانی رخ می‌دهد که عملیات نوشتن در یک سلول، به طور ناخواسته بر سلول‌های مجاور تأثیر می‌گذارد. این مسئله به‌ویژه در حافظه‌های TLC و MLC که از چندین سطح ولتاژ برای ذخیره‌سازی داده‌ها استفاده می‌کنند، بحرانی‌تر است. از سوی دیگر، تداخل سلول به سلول ناشی از اثرات خازنی بین سلول‌های مجاور است و با کوچک‌تر شدن ابعاد فیزیکی سلول‌ها، این مشکل تشدید می‌شود.



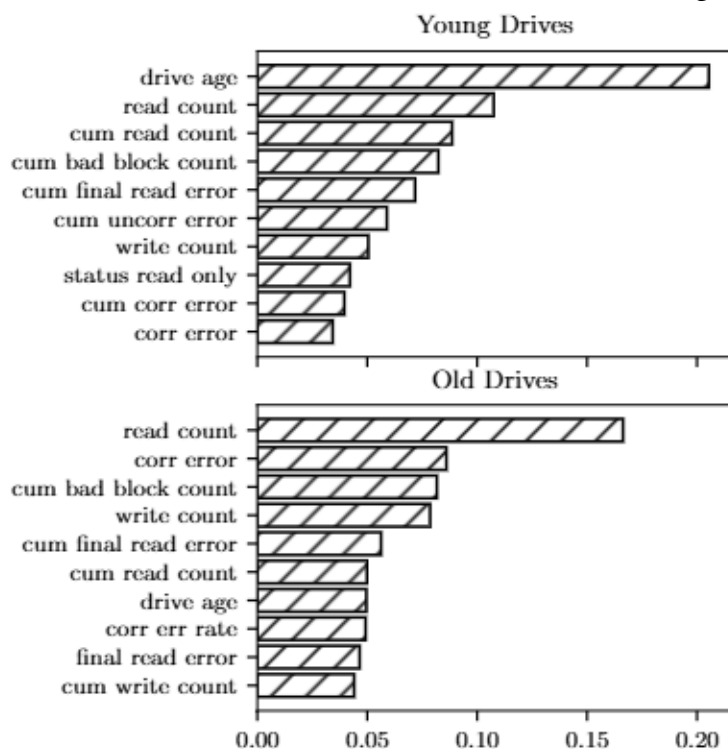
شکل ۲: نشان دادن تاثیر خواندن و نوشتن بر روی سلول های مجاور در درایو

یکی از دیگر مشکلات خطاهای نگهداری و همچنین اختلال در خواندن از دیسک است که خطای نگهداری به علت نشت تدریجی شارژ الکتریکی از گیت شناور سلول به وجود می آید. که یکی از عوامل تشدید این مشکل افزایش دما است که میتواند منجر به از دست دادن اطلاعات شود. اختلال در خواندن زمانی ایجاد میشود که خواندن مکرر از یک بلوک حافظه باعث تغییر تدریجی در وضعیت سلول های مجاور میشود. دیگر عامل ایجاد خطا وجود بلوک معیوب در زمان تولید و یا به وجود آمدن آن در طول عمر دیسک است. که سیستم مدیریت دیسک باید بتواند این بلوک ها را شناسایی کند و تا از اختلال در سیستم جلوگیری کند

تجميع اطلاعات مقالات علمی مشابه با موضوع پروژه

یکی از مواردی که در مقالات مشابه مشاهده شده است. مجموعه داده های آن هاست که به خاطر استفاده از درایو های متفاوتی مجموعه داده هایی بیشتر از ویژگی های استاندارد SMART در خود دارند که شرکت سازنده به محصولات خود اضافه کرده است. مانند دمای درایو در اکثر پژوهش ها الگوهای خرابی دیسک بیشتر بر اساس معیارهایی همچون چرخه های نوشتن/پاک کردن، دما و تعداد عملیات خواندن و نوشتن بررسی شده است. اما نتایج مقاله ی مورد بررسی نشان می دهد که نرخ خرابی درایو ها در دوران نوزادی، که مقاله این دوران را ۹۰ روز اول ذکر کرده است نسبت به دیگر دوران ها بالاتر است (قابل توجه هست که ما در بررسی های اولیه در پروژه خود نیز عاملی خرابی درایو ها در دوران نوزادی را میبینیم که در بخش های بعد آورده شده است). همچنین مورد دیگری که مقاله اشاره میکند آن است که چرخه های نوشتن و پاک کردن به اندازه ایی که تصور میشده در پیش بینی خرابی ها موثر نبوده اند. به طور کلی این مقاله نشان میدهد که خرابی ها در درایو ها بیشتر به عوامل متنوع وابسته هستند که نمیتوان آن ها را به یک عامل منفرد نسبت داد. این

مقاله از یادگیری ماشین برای پیش بینی خرابی درایو استفاده کرده است که مدل پیشنهادی این مقاله استفاده از الگوریتم جنگل تصادفی است.



شکل ۳: بررسی اهمیت عوامل تاثیر گذار بر خرابی درایو ها بر اساس سن آن ها به دست آمده توسط روش جنگل تصادفی

۱. عوامل مرتبط با سن و عمر دستگاه:

- دوره نوزادی (۹۰ روز اول): در این دوره احتمال خرابی بسیار بالاست.
- عمر کلی دستگاه: پس از دوره نوزادی، تاثیر سن بر خرابی کمتر می شود.
- تعداد چرخه های نوشتن و پاک کردن (P/E Cycles).

۲. خطاهای عملیاتی:

- خطاهای قابل تصحیح (توسط سیستم ECC).
- خطاهای غیر قابل تصحیح.
- خطاهای پاک کردن.
- خطاهای نهایی خواندن و نوشتن.
- خطاهای مربوط به متادیتا.

- خطاهای پاسخ‌دهی سیستم.
- خطاهای وقفه زمانی.

۳. عملکرد بلوک‌ها:

- تعداد بلوک‌های معیوب کارخانه‌ای.
- تعداد بلوک‌های معیوب ایجاد شده در حین کار.
- نرخ افزایش بلوک‌های معیوب.

۴. الگوهای استفاده:

- تعداد عملیات خواندن روزانه.
- تعداد عملیات نوشتن روزانه.
- تعداد عملیات پاک کردن.
- حجم کلی داده‌های پردازش شده.

۵. وضعیت‌های سیستمی:

- حالت فقط-خواندنی.
- وضعیت مرگ درایو.
- زمان پاسخ‌دهی سیستم.

۶. عوامل پیش‌بینی کننده در درایوهای جوان (کمتر از ۹۰ روز):

- سن دستگاه.
- تعداد تجمعی بلوک‌های معیوب.
- خطاهای نهایی خواندن.
- تعداد خطاهای خواندن.
- خطاهای غیرقابل تصحیح.

۷. عوامل پیش‌بینی کننده در درایوهای بالغ (بیشتر از ۹۰ روز):

- تعداد عملیات خواندن.

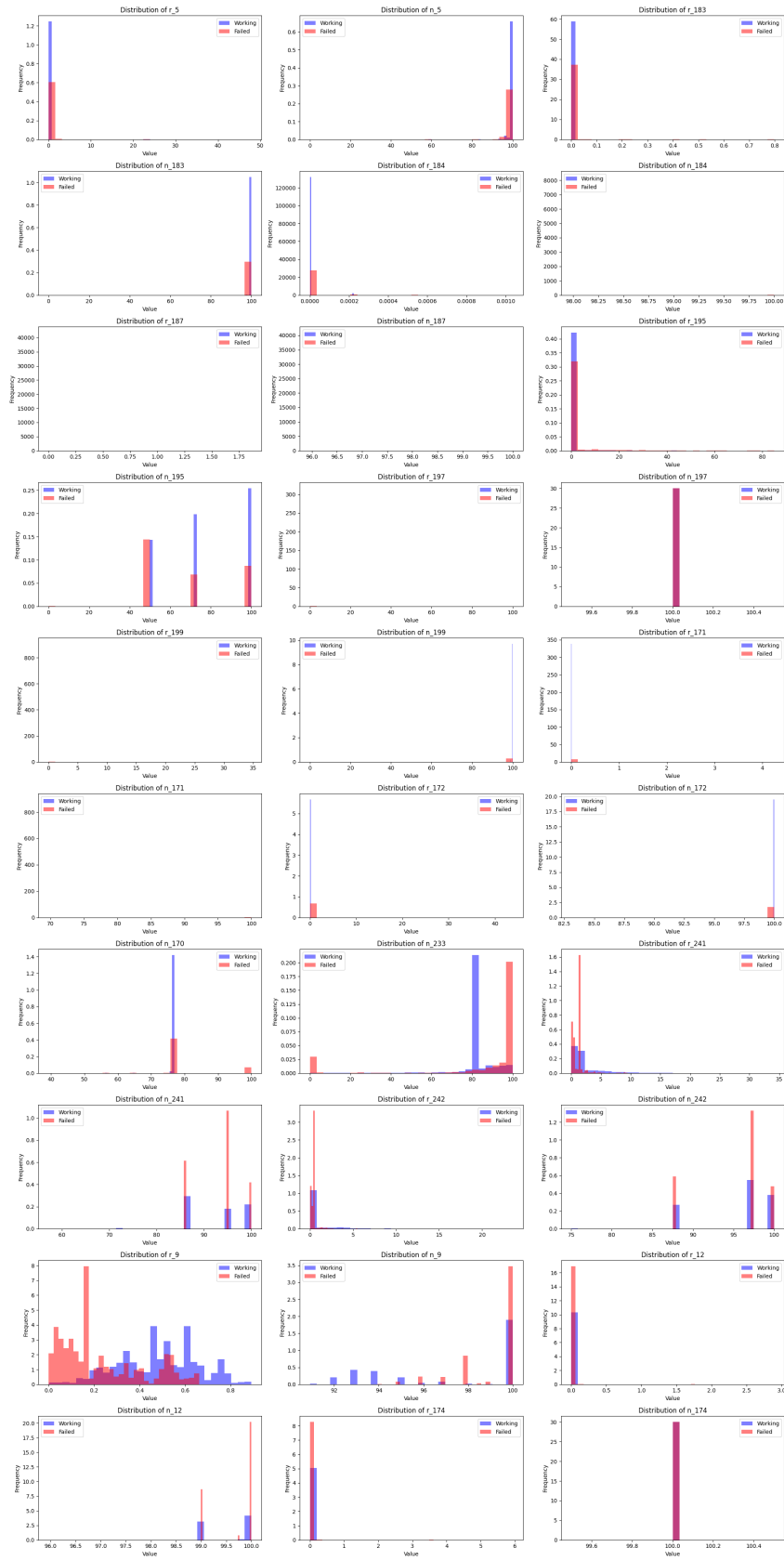
- خطاهای قابل تصحیح.
- تعداد عملیات نوشتن.
- تعداد تجمعی بلوک‌های معیوب.

مقاله دیگری که مورد بررسی قرار دادیم این نکته را مورد توجه قرار می‌دهد که بسیاری از روش‌های موجود برای پیش‌بینی خرابی درایوها تنها بر داده‌های کوتاه مدت تاکید دارند و نمیتوانند الگوهای بلند مدتی که ممکن است خرابی‌های قریب الوقوع را تشخیص دهند پیدا کنند. روشی که این مقاله پیشنهاد می‌دهد روش MVTRF (Multi-View and Multi-Task Random Forest) نام دارد که این روش از سه ویژگی کلیدی برای پیش‌بینی خرابی‌ها استفاده میکند که هر کدام می‌تواند اطلاعات متفاوتی را در اختیار مدل قرار دهد. ویژگی اول ویژگی‌های خام SMART هستند. ویژگی بعدی ویژگی‌های هیستوگرام هستند که سعی می‌کنند توزیع‌های آماری اندازه‌گیری‌ها را در طول مدت طولانی تجزیه و تحلیل کنند و به شناسایی روندهای بلند مدت کمک می‌کنند. و ویژگی سوم ویژگی‌های مرتبط با توالی است که سعی می‌کند روندی که داخل داده‌ها موجود است را استخراج کند. با توجه به این مقاله ما هم سعی کرده‌ایم که در انجام پروژه بتوانیم ویژگی‌های و الگوهای بلند مدت تر را مورد توجه قرار دهیم.

۳-۰ پیش پردازش داده و کسب اطلاعات در مورد آن‌ها

ابتدا سعی میکنیم با توجه به اطلاعاتی که از مرور مقالات گذشته به دست آورده ایم اطلاعاتی از نحوه توزیع داده‌ها به دست آوریم با توجه به خرابی آن‌ها. در مجموعه داده موجود ما که دارای ۵۰۰۰ سطر است که در واقع اطلاعات SMART داده‌های درایو‌ها هستند که در هر ستون قرار گرفته‌اند ابتدا در تصویر زیر نمودارهایی از توزیع هر یک از ستون‌ها با توجه به حالت خرابی آن‌ها را می‌بینیم.

Distribution of Features in Training Data Set (Histogram)



شکل ۴: هیستوگرام توزیع مقادیر هر یک از ستون ها نسبت با توجه به خراب یا سالم بودن درایوها

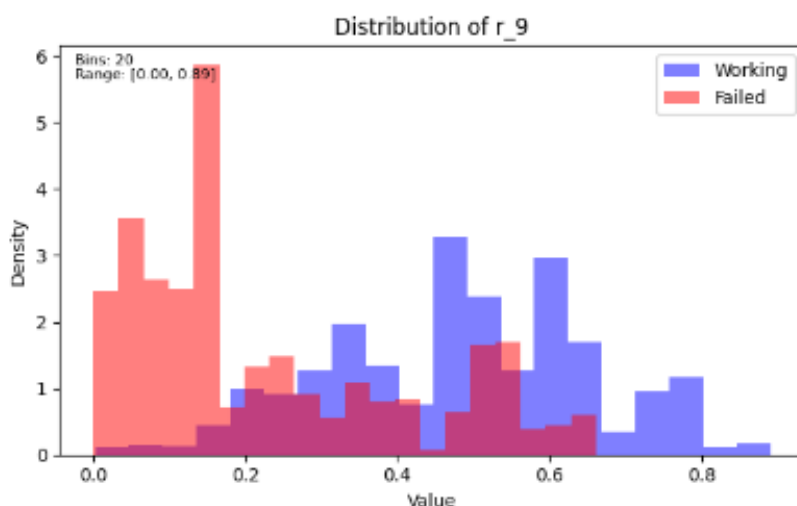
در شکل آن هایی که با قرمز مشخص شده اند درایو هایی را نشان میدهد که دچار خرابی شده اند و آبی ها دیسک های سالم را نشان میدهد. اولین نکته ایی که توجه ما را جلب میکند آن است که در بعضی از ستون ها میتوان مرز های مشخصی برای جدا کردن درایو های خراب از سالم در نظر گرفت در حالی که در بعضی دیگر مرز مشخصی وجود ندارد و چه بسا دو هستوگرام کاملاً بر روی هم افتاده باشند.

در تصویر زیر توضیحاتی در مورد اینکه هر ستون مربوط به چه اطلاعاتی است میبینید.

Item	Attr.	Description	Item	Attr.	Description
0	Id	id	16	R171	Program Failed Count
1	diskid	Disk ID	17	N171	Program Failed Count
2	R5	Reallocated Sector Count	18	R172	Erase Failed Count
3	N5	Reallocated Sector Count	19	N172	Erase Failed Count
4	R183	SATA Downshift Error	20	N170	Available Reserved Blocks
5	N183	SATA Downshift Error	21	N233	Media Wearout Indicator
6	R184	End-to-End Errors	22	R241	Number of Blocks Written
7	N184	End-to-End Errors	23	N241	Number of Blocks Written
8	R187	Reported Uncorrectable Errors	24	R242	Number of Blocks Read
9	N187	Reported Uncorrectable Errors	25	N242	Number of Blocks Read
10	R195	Hardware ECC Recovered	26	R9	Power on Hours
11	N195	Hardware ECC Recovered	27	N9	Power on Hours
12	R197	Current Pending Sector Count	28	R12	Power Cycle Count
13	N197	Current Pending Sector Count	29	N12	Power Cycle Count
14	R199	Ultra DMA CRC Error Count	30	R174	Unexpected Power Loss Count
15	N199	Ultra DMA CRC Error Count	31	N174	Unexpected Power Loss Count

شکل ۵: نام دقیق هر ستون در مجموعه داده

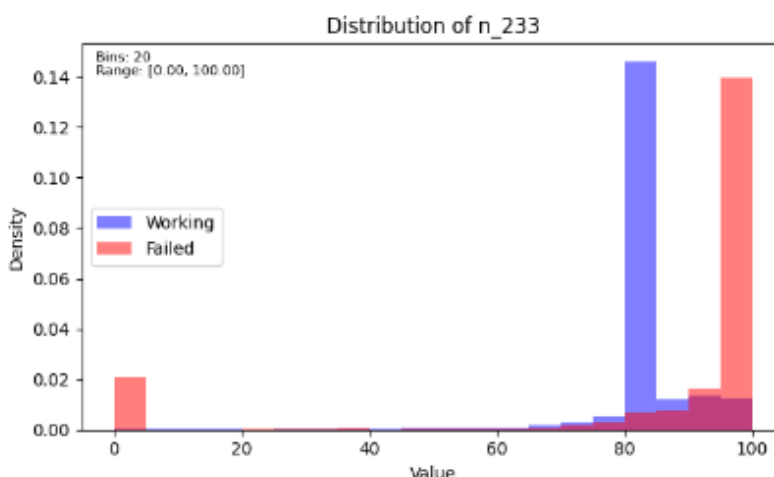
حال به توضیح ستون هایی میپردازیم که هستوگرام داده های درایوهای خراب و سالم قابل جدا کردن است



شکل ۶: هستوگرام ستون داده خام ۹

این ستون که با توجه به جدول مربوط به مدت زمان روشن بودن دیسک ها است نکته قابل توجهی که مشاهده میشود که در مقالاتی که در بخش قبل آورده بودیم همه ذکر شده بود این نکته

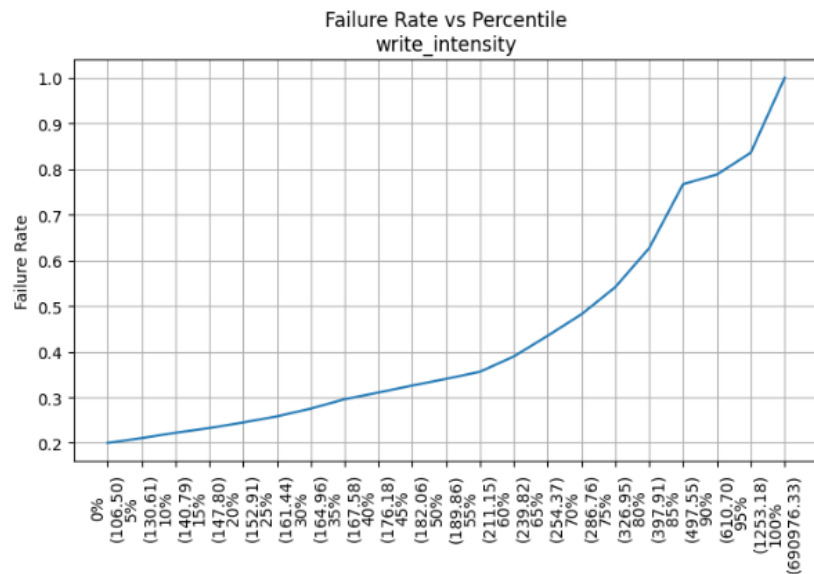
است که بخش زیادی از دیسک هایی که خراب شده اند در ابتدای طول عمر خود بوده اند.



شکل ۷: هیستوگرام ستون نرمالایز شده ۲۳۳

این ستون که مربوط به اندیکاتور فرسودگی درایو است. قابل مشاهده است که درایوهای خراب و سالم قابل جدا سازی هستند و هر چه این اندیکاتور مقدار بیشتری را نشان دهد نشان دهنده این است که احتمال خرابی سیستم بیشتر است.

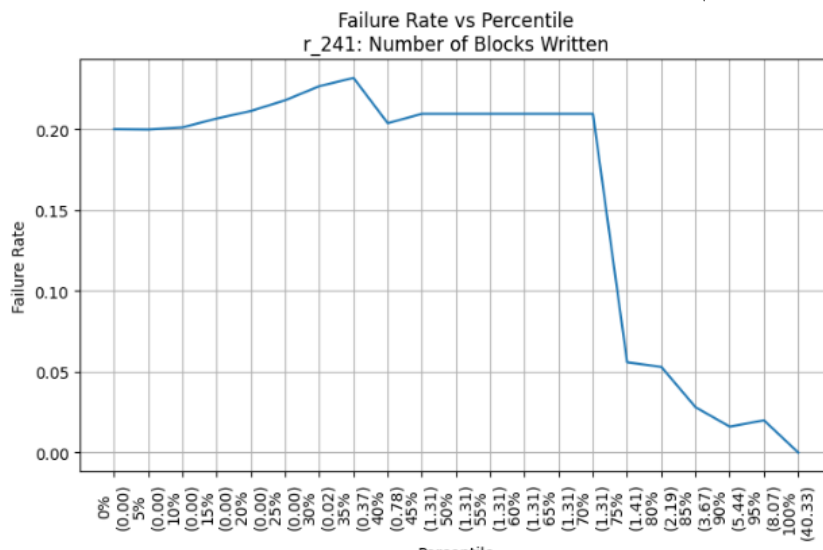
نکته ایی که تا اینجا مورد توجه قرار گرفته است این است که ستون هایی که با استفاده از آن ها بتوان مشخص کرد که آیا درایو میتواند خراب باشد یا خیر بسیار کم است پس سعی میکنیم با استفاده از مهندسی ویژگی های اسمارت ویژگی های جدیدی برای خود به وجود آوریم. همانطور که در مقالات گذشته بررسی شده بود و خودمان میدانیم این است که یکی از عوامل تاثیر گذار بر خرابی درایو تعداد عملیات های نوشتن بر آن است که در ستون های داده های ما این مورد موجود است اما همانطور که در هیستوگرام ها دیدیم دیسک های خراب و سالم با استفاده از این موضوع از هم قابل جدا سازی نبودند بنابراین به ویژگی بهتری نیاز داریم که خرابی های درایو ها را بتواند بهتر به ما نشان دهد. اگر میزان نوشتنی که بر روی هر دیسک انجام شده است را بر مدت زمانی که دیسک روشن بوده است تقسیم کنیم میتوانیم ویژگی جدیدی که نشان دهنده میزان نوشتن ها با توجه به طول عمر دیسک است را به دست آوریم که در نمودار زیر میتوان دید که چگونه این ویژگی جدید میتواند درایو های خراب را از سالم جدا کند



شکل ۸: نمودار ویژگی مهندسی شده شدت نوشتن

$$\text{write_intensity} = \frac{\text{Written Blocks of Number : ۲۴۱n}}{\text{Hours on Power : ۹r}}$$

همانطور که مشاهده میکنید نمودار در محور افقی مقادیر فیچر جدید بر اساس درصد آن ها از کل داده ها را نشان میدهد و در محور عمودی نرخ خرابی را نشان میدهد که با افزایش چگالی نوشتن باید نرخ خرابی هم افزایش یابد



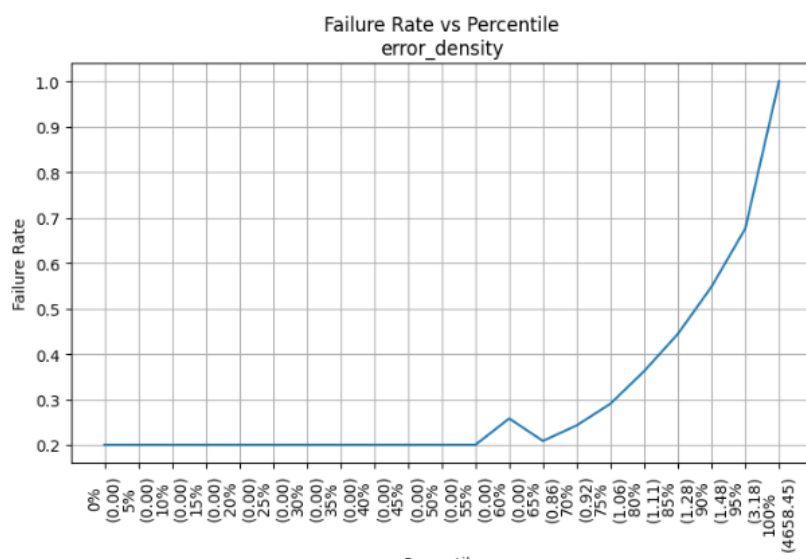
شکل ۹: نمودار ویژگی تعداد نوشتن

$$\text{read_write_ratio} = \frac{\text{Read Blocks of Number : ۲۴۲n}}{\text{Written Blocks of Number : ۲۴۱n}}$$

همانطور که در نمودار بالا مشاهده میشود تفاوت معنا داری نمیتوان از روی تعداد نوشتن برای یافتن خرابی ها مشاهده کرد

ویژگی های مهندسی شده دیگر که به ویژگی های اصلی مجموعه داده اضافه شده است دیگر نسبت تعداد کل خواندن ها به نوشتن ها است که از تقسیم تعداد خواندن ها به نوشتن ها به دست آمده است. و ویژگی دیگر هم چگالی خطا ها است که از مجموع ستون خطاهای غیر قابل تصحیح کردن، تعداد خطاهای تصحیح شده به وسیله کد تصحیح کننده و خطاهای مربوط به انتقال داده از درایو به دی ام ای است تقسیم بر مدت زمان روشن بودن درایو که نمودار این ویژگی هم در زیر

میبینید



شکل ۱۰: نمودار ویژگی تعداد چگالی خطاها

$$\text{error_density} = \frac{(\text{r_187: Reported Uncorrectable Errors} + \text{r_195: Hardware ECC Recovered} + \text{r_199: Ultra DMA CRC Error Count})}{\text{r_9: Power on Hours}}$$

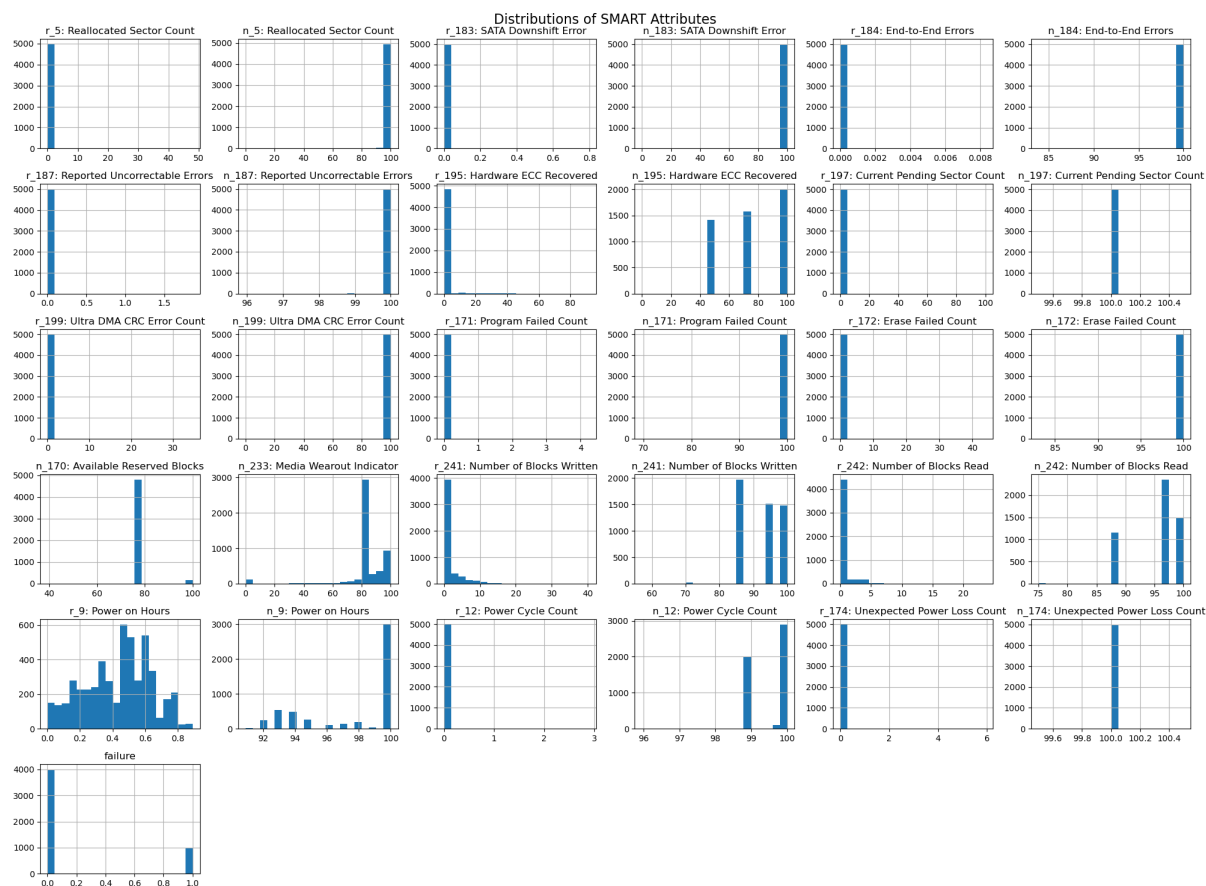
۴-۰ پیاده‌سازی

۱-۴-۰ بخش اول: پیش‌بینی زمان خرابی دیسک

برای کشف زمان خرابی دیسک‌ها ما چنین برداشت کردیم که بایستی زمان پارامتر Power on Hour را برای دیسک‌ها محاسبه کنیم و در همین راستا پس از خواندن داده‌ها از فایل داده‌ی قرار گرفته در اختیارمان، داده‌ها را در ابتدا تمیز کرده و ستون‌های داده‌ها را مطابق با شکل یک تغییر نام دادیم که در شکل ۱۰ نام‌های ستون‌ها آمده است؛ سپس توزیع داده‌ها و همچنین نحوه‌ی توزیع آن‌ها بر اساس پارامتر مذکور را بدست آوردیم که در شکل‌های ۱۲ و ۱۳ نشان داده شده‌اند.

Item	Attr.	Description	Item	Attr.	Description
0	Id	id	16	R171	Program Failed Count
1	diskid	Disk ID	17	N171	Program Failed Count
2	R5	Reallocated Sector Count	18	R172	Erase Failed Count
3	N5	Reallocated Sector Count	19	N172	Erase Failed Count
4	R183	SATA Downshift Error	20	N170	Available Reserved Blocks
5	N183	SATA Downshift Error	21	N233	Media Wearout Indicator
6	R184	End-to-End Errors	22	R241	Number of Blocks Written
7	N184	End-to-End Errors	23	N241	Number of Blocks Written
8	R187	Reported Uncorrectable Errors	24	R242	Number of Blocks Read
9	N187	Reported Uncorrectable Errors	25	N242	Number of Blocks Read
10	R195	Hardware ECC Recovered	26	R9	Power on Hours
11	N195	Hardware ECC Recovered	27	N9	Power on Hours
12	R197	Current Pending Sector Count	28	R12	Power Cycle Count
13	N197	Current Pending Sector Count	29	N12	Power Cycle Count
14	R199	Ultra DMA CRC Error Count	30	R174	Unexpected Power Loss Count
15	N199	Ultra DMA CRC Error Count	31	N174	Unexpected Power Loss Count

پس از مشاهده‌ی نحوه‌ی توزیع داده‌ها، تصمیم گرفتیم که یک مدل regression را بر داده‌های مذکور فیت کنیم؛ در ادامه مدل‌های مختلفی که امتحان کردیم و نتایجی که بدست آوردیم را گزارش خواهیم کرد. دلیل استفاده از مدل regression این بود که نیازمند تخمین مقدار بودیم پس به یک تابع نیاز داریم که با توجه به پارامترهای ورودی (مجموعه featureها) پارامتر مورد نیاز ما (پارامتر target) را محاسبه کند.

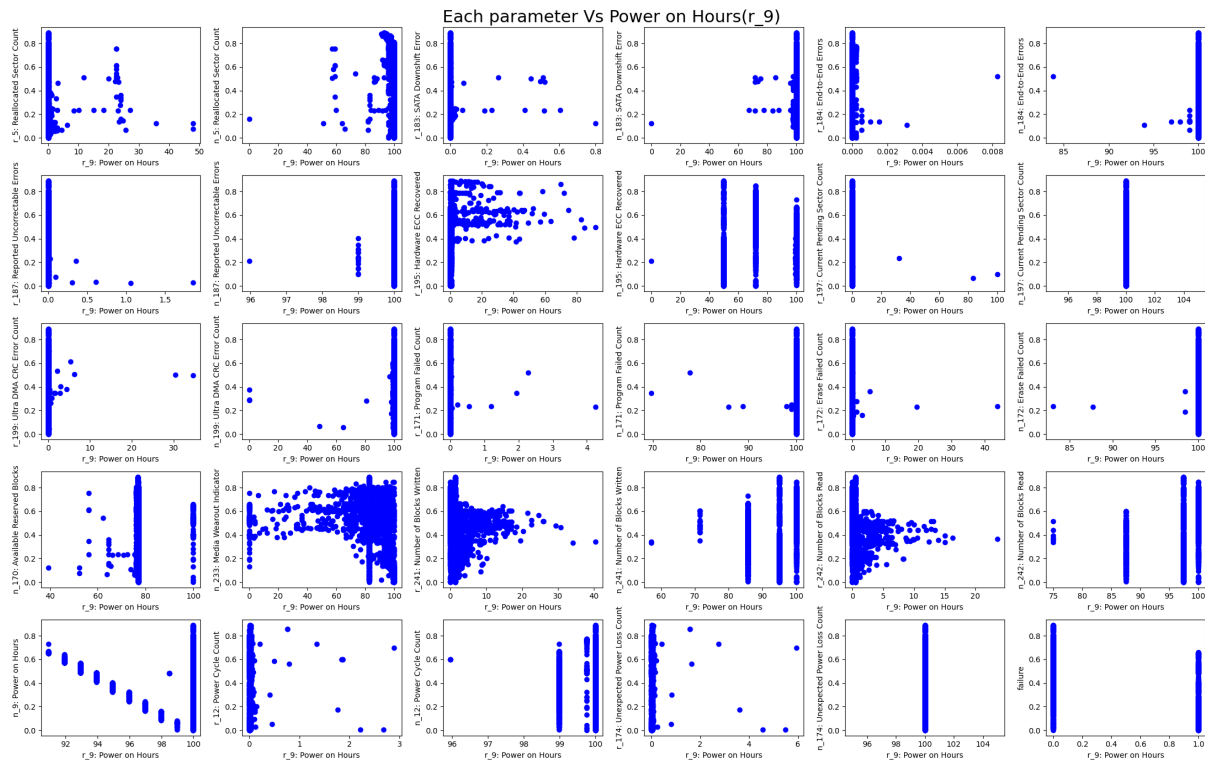


Regression (رگرسیون)

رگرسیون یکی از تکنیک‌های مهم در یادگیری ماشین است که برای پیش‌بینی مقادیر پیوسته یا عددی بر اساس داده‌های ورودی استفاده می‌شود. در رگرسیون، هدف مدل‌سازی و یافتن روابط میان متغیر وابسته (یا هدف) و متغیرهای مستقل است تا با استفاده از این روابط، مقادیر جدید برای متغیر وابسته پیش‌بینی شود. در واقع، رگرسیون به عنوان ابزاری برای برآورد مقدار یک متغیر بر اساس سایر متغیرها شناخته می‌شود. به عبارت دیگر، مدل‌های رگرسیون به ما این امکان را می‌دهند که روند و الگوهای موجود در داده‌ها را شبیه‌سازی کرده و از آن برای پیش‌بینی نتایج استفاده کنیم.

۱. Linear Regression (رگرسیون خطی)

با توجه به توضیحات ارائه شده در ارائه‌ی اول ما از مدل‌های Linear Regression، Random Forest Regression، Support Vector Regression و XGBoost Regressor به عنوان مدل‌های پایه استفاده کرده بودیم و فرآیند تشخیص خطا را تنها بر روی دیسک‌های خراب انجام داده بودیم. در ادامه به توضیح هر یک می‌پردازیم و نتیجه‌ی بدست آمده



را گزارش می‌کنیم. در ادامه نیز با استفاده از یک مدل شبکه عصبی عمیق (DNN) بهره بردیم که در خروجی به ما پارامترهای توزیع احتمال یا به نوعی همان توزیع احتمال خرابی را می‌دهد. همچنین علاوه بر این نیز از Cox Proportional Hazards بهره بردیم که یک مدل نیمه‌پارامتری است و با استفاده از مجموعه داده‌ی دیسک‌ها خراب و دیسک‌های تاکنون خراب نشده نمودار خرابی دیسک‌ها Baseline Hazard را بر اساس partial likelihood بدست می‌آورد. در ادامه به توضیح این موارد پرداخته و نتایج بدست آمده را نیز گزارش می‌کنیم. Linear Regression از ساده‌ترین و رایج‌ترین روش‌های رگرسیون که فرض می‌کند رابطه‌ای خطی میان متغیرهای مستقل و وابسته وجود دارد. در این مدل، هدف یافتن یک خط است که بهترین برازش را روی داده‌ها داشته باشد. به عبارت دیگر، مدل linear regression سعی می‌کند معادله‌ای به شکل $y = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$ پیدا کند که در آن y متغیر وابسته و x_1, x_2, \dots, x_n متغیرهای مستقل هستند و $\beta_0, \beta_1, \dots, \beta_n$ ضرایب مدل هستند که باید به‌طور بهینه تعیین شوند. این روش معمولاً زمانی کاربرد دارد که رابطه‌ای خطی بین متغیرها وجود داشته باشد و داده‌ها به‌طور مستقیم و با شیب ثابت تغییر کنند.

۲. Random Forest Regressor (رگرسیون جنگل تصادفی)

Random Forest Regressor یکی از روش‌های مبتنی بر یادگیری جمعی است که از ترکیب چندین درخت تصمیم‌گیری برای پیش‌بینی استفاده می‌کند. در این روش، ابتدا چندین درخت تصمیم‌گیری مستقل ساخته می‌شود و هر یک پیش‌بینی خود را انجام می‌دهد. سپس، نتایج پیش‌بینی‌ها با یکدیگر ترکیب شده و پیش‌بینی نهایی به‌عنوان میانگین یا غالب‌ترین پیش‌بینی درخت‌ها انتخاب می‌شود. این روش قادر است با استفاده از نمونه‌برداری تصادفی ویژگی‌ها و داده‌ها، تعمیم‌پذیری بالاتری به مدل دهد و به‌طور مؤثر از مشکل اورفیتینگ (overfitting) جلوگیری کند. به دلیل ویژگی‌های آن، Random Forest برای داده‌های پیچیده و بزرگ مناسب است و می‌تواند روابط غیرخطی پیچیده را نیز مدل‌سازی کند.

۳. (SVR) Support Vector Regression (رگرسیون ماشین بردار پشتیبانی)

رگرسیون ماشین بردار پشتیبانی (SVR) یک روش قدرتمند در یادگیری ماشین است که برای مسائل رگرسیون استفاده می‌شود. برخلاف رگرسیون خطی که به دنبال یک خط برای بهترین برازش است، SVR به دنبال یک «صفحه» (در ابعاد بیشتر از ۲) است که بتواند به‌طور مؤثر داده‌ها را پیش‌بینی کند. SVR به‌طور خاص به کاهش خطاها و پیش‌بینی‌های دقیق‌تر با محدود کردن اندازه اشتباهات به کمک استفاده از یک کرنل مناسب پرداخته و به دنبال کمینه کردن خطای پیش‌بینی در محدوده‌ای مشخص از داده‌ها می‌باشد. یکی از ویژگی‌های برجسته SVR، توانایی آن در مدل‌سازی روابط غیرخطی و استفاده از تکنیک‌هایی مانند کرنل‌سازی برای تبدیل داده‌ها به فضاها با ابعاد بالاتر است تا مدل به راحتی بتواند مرزهای پیچیده را شبیه‌سازی کند.

۴. XGBoost Regressor (رگرسیون XGBoost)

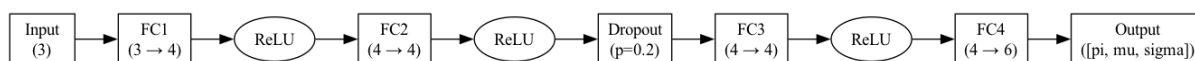
XGBoost (Extreme Gradient Boosting) یکی از الگوریتم‌های قدرتمند و محبوب در یادگیری ماشین است که برای حل مسائل رگرسیون و طبقه‌بندی استفاده می‌شود. این الگوریتم مبتنی بر روش تقویت گرادیان (Gradient Boosting) است که در آن مدل‌ها به‌طور مرحله‌ای ساخته می‌شوند و هر مدل جدید سعی می‌کند خطاهای مدل قبلی را جبران کند. XGBoost به‌ویژه در رقابت‌های علم داده به دلیل دقت بالا، سرعت و توانایی مقابله با داده‌های بزرگ و پیچیده شناخته شده است. یکی از ویژگی‌های منحصر به‌فرد XGBoost، استفاده از تکنیک‌هایی مانند کاهش بیش‌برازش (regularization) و بهینه‌سازی دقیق در فرآیند یادگیری است که موجب بهبود عملکرد مدل در مسائل پیچیده می‌شود.

مطابق با توضیحات داده شده، ما هر ۴ الگوریتم یاد شده به همراه الگوریتم رگرسیون چندجمله‌ای را بر روی داده‌های دیسک‌های خراب شده به منظور کشف تابع زمان خرابی اجرا کردیم که با ۸۰ درصد داده‌ها آموزش دید و بر روی ۲۰ درصد مابقی داده‌ها تست شد. بدین صورت الگوریتم جنگل تصادفی مطابق با انتظار از پیش تعریف شده‌ی ما با دقت ۹۱ درصد بر روی مجموعه‌ی داده‌های تست بیشترین دقت را داشت که از سویی نیز این دقت به میزان کافی بر اساس سائز مجموعه داده و توزیع مقادیر آن خوب است. هنگامی که از رگرسیون خطی ساده استفاده کردیم داده‌ها باعث می‌شدند دقت بسیار پایینی بدست بیاوریم به همین منظور با استفاده از SelectKBest ده تا از موثرترین فیچرها را برای رگرسیون انتخاب کردیم که میزان همبستگی خوبی با پارامتر Power on Hour داشتند اما باز هم تابع بدست آمده بهترین تابع نبود و دقت معادل ۷۰ درصد داشت. در نهایت بهتر است که یکبار هم با استفاده از شبکه‌ی عصبی عمیق و همچنین شبکه‌های یادگیری بازگشتی مسئله حل شود چراکه با حل کنونی ما تنها از داده‌های دیسک‌های خراب استفاده کردیم ولی اگر راهکارهای یاد شده را پیش بگیریم امکان استفاده از داده‌های دیسک‌های سالم در فرآیند یادگیری را نیز می‌توانیم داشته باشیم.

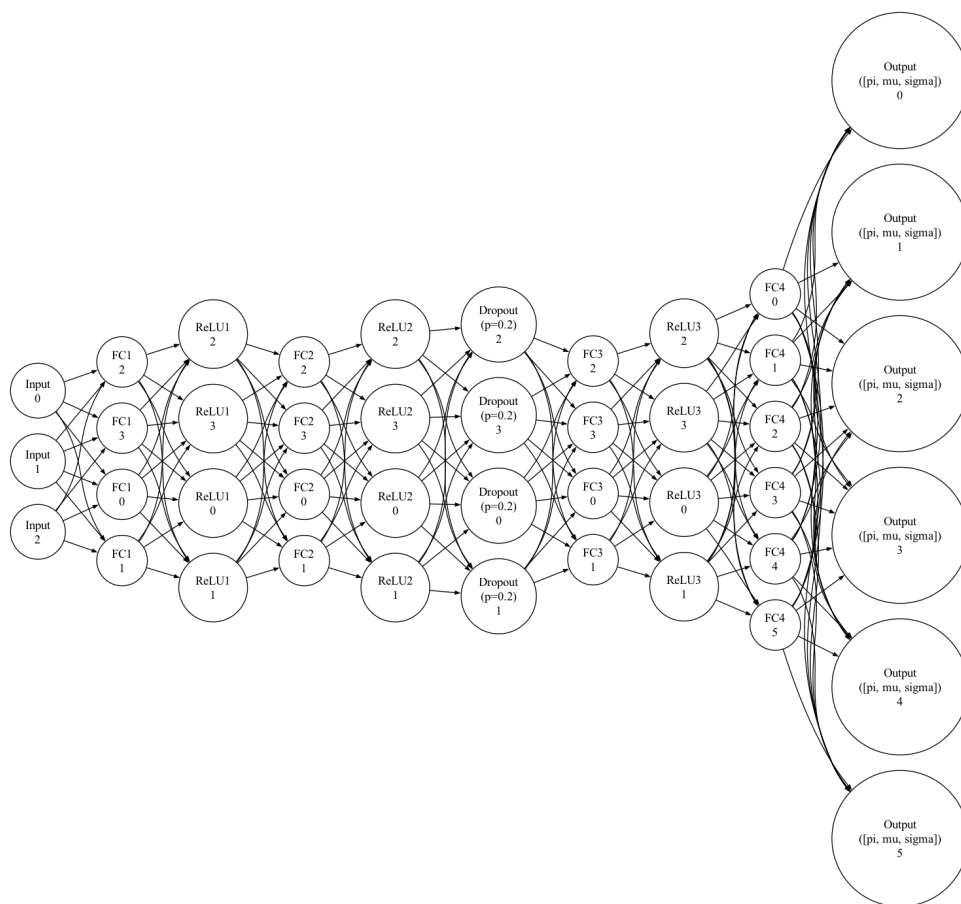
۱. روش اول: شبکه‌ی عصبی عمیق

در این قسمت سعی کردیم با استفاده از شبکه عصبی بتوانیم توزیع احتمال ساعت روشن بودن دیسک‌ها را به دست آوریم برای این کار به دست آوردن توزیع احتمال روشن بودن دستگاه‌ها کاری که انجام دادیم این بود که چون شبکه عصبی نمیتواند خروجی مستقیم توزیع احتمال را بر اساس ورودی دهد از روش ساخت توزیع احتمال از روی توزیع احتمال نرمال استفاده کردیم که فرمول آن در زیر آمده است

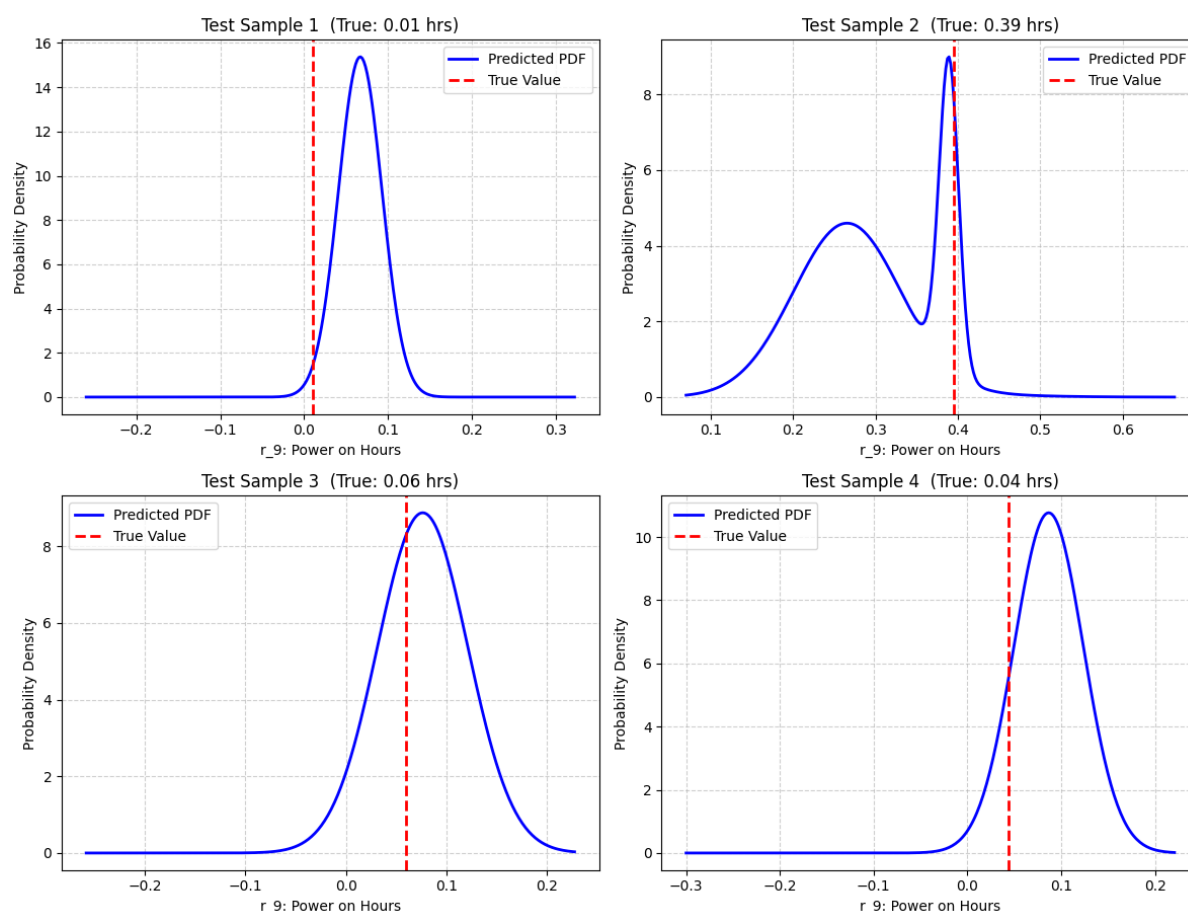
$$p(y | x) = \sum_{i=1}^{\text{num_mixtures}} \pi_i \cdot \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{y - \mu_i}{\sigma_i}\right)^2\right)$$



که شبکه عصبی با توجه به دقتی که می‌خواهیم تعداد توزیع‌های گوسی را انتخاب کنیم و شبکه برای هر توزیع یک پی و میانگین و واریانس را به ما خروجی میدهد که پس از اجرا کردن این مدل بر روی داده‌ها نتایج زیر را می‌بینیم



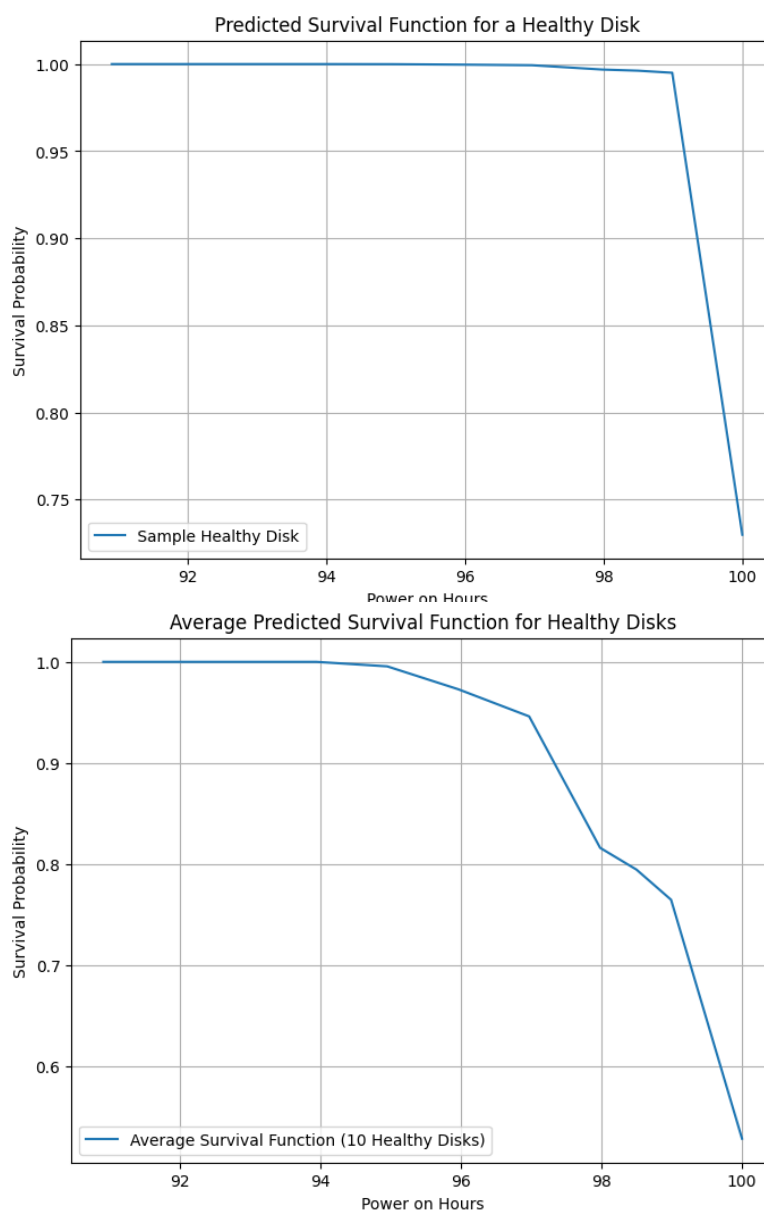
Predicted PDF and True Power on Hours for Four Random Test Samples



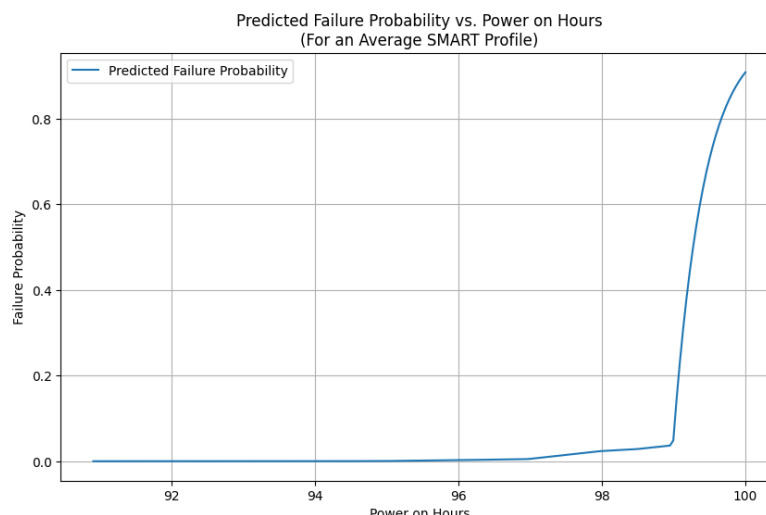
۲. روش دوم: استفاده از Cox Proportional Hazards

در این بخش با استفاده از Cox Proportional Hazards، عملکرد دیسک‌ها را بر اساس ساعت کاری مورد بررسی قرار دادیم. این مدل نیمه‌پارامتری، بدون نیاز به فرض تابع بقا پایه، با بهره‌گیری از اطلاعات دیسک‌هایی که فیل شده‌اند (رخداد) و دیسک‌هایی که هنوز خراب نشده‌اند (داده‌های سانسور شده)، اثر متغیرهای توضیحی مانند شاخص‌های SMART را بر روی خطر فیلیر برآورد می‌کند. به عبارت دیگر، مدل با استفاده از مفهوم partial likelihood در هر زمان رخداد، مجموعه‌ای از دیسک‌های در معرض خطر را بررسی کرده و ضرایب مربوط به متغیرها را برآورد می‌کند. در بخش‌های مختلف کد، ابتدا تابع بقا برای یک دیسک سالم نمونه و همچنین گروهی از دیسک‌های سالم پیش‌بینی شده و رسم می‌شود، که نشان می‌دهد با افزایش ساعت کاری احتمال سالم ماندن دیسک کاهش می‌یابد. سپس با استفاده از پروفایل میانگین به دست آمده از مقادیر شاخص‌های SMART، تابع بقا و به تبع آن احتمال فیلیر (معادل ۱ منهای تابع بقا) نیز پیش‌بینی و نمودار آن رسم می‌شود. این روند پیش‌بینی به وضوح نشان می‌دهد که هرچه ساعت

کاری افزایش یابد، احتمال وقوع خرابی نیز به صورت افزایشی تغییر می‌کند. نتایج به دست آمده در نمودارها کاملاً منطقی و مطابق انتظار هستند؛ یعنی نمودار تابع بقا نشان‌دهنده کاهش تدریجی احتمال زنده ماندن دیسک‌ها در طول زمان و نمودار احتمال فیلیر نیز روند افزایشی با گذشت ساعت‌های کاری را به نمایش می‌گذارد. همچنین نتایج حاصل از تحلیل گروهی و پروفایل میانگین، با نتایج فردی به خوبی همخوانی داشته و تصویری جامع از رفتار دیسک‌ها ارائه می‌دهد.

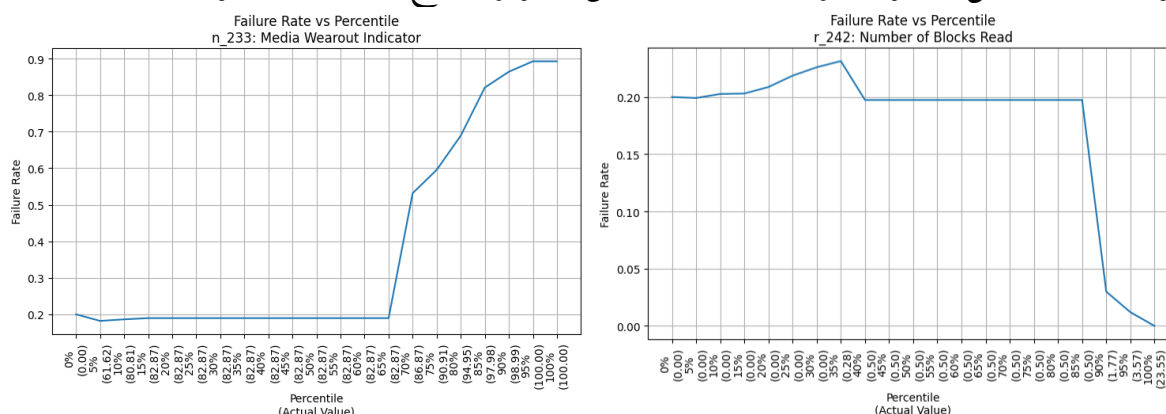


همانطور که در تصاویر اول و سوم مشخص است این دو نمودار رفتاری عکس یکدیگر دارند که منطقی است همچنین توجیح رفتاری نمودار سوم با توضیحات ارائه شده در بخش دوم در ادامه توضیح داده می‌شود. در ادامه نیز برای حل بخش سوم از همان شبکه عصبی عمیق بخش اول استفاده شده است و از راهکار دوم توضیح داده شده استفاده نکردیم.

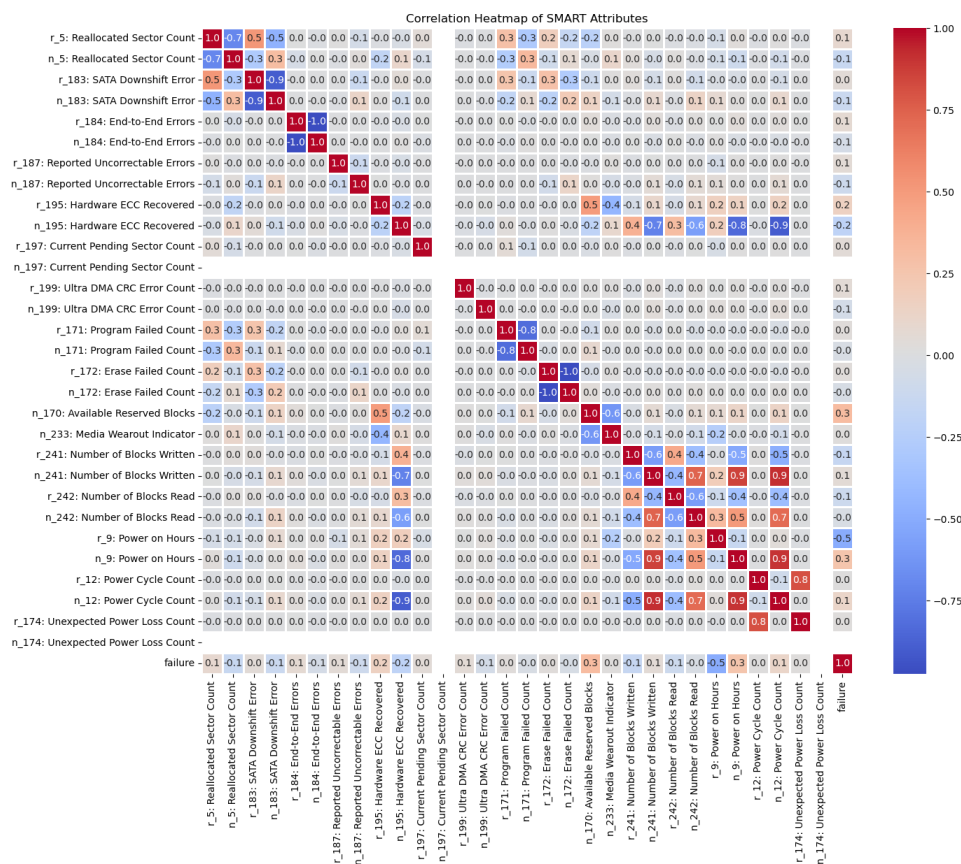


۲-۴-۰ بخش دوم: محاسبه مقادیر آستانه برای ویژگی‌های مختلف اسمارت:

در این قسمت برای محاسبه‌ی مقدار آستانه‌ی هر یک از ویژگی‌های اسمارت، ما در ابتدا نمودار Failure Rate بر اساس درصد رخداد هر یک از پارامترها را رسم نمودیم. در اینصورت در هر یک از نقاط رو به رشد توابع رسم شده می‌توان چنین نتیجه گرفت که آن نقطه، مقدار آستانه برای ویژگی مذکور می‌تواند باشد چراکه پس از آن میزان خرابی به طور سعودی افزایش خواهد یافت. به عنوان مثال به شکل ۱۴ توجه فرمایید. مطابق با این تصویر واضح است که به عنوان مثال حد آستانه



برای پارامتر Media Wearout indicator یا همان n 233 برابر حدودا ۷۰ درصد است چرا که بعد از اینکه این پارامتر به ۷۰ درصد خود برسد روند خرابی دیسک به طور سعودی افزایش خواهد یافت. همچنین برای ارتباط و همبستگی هر یک از پارامترها با خرابی دیسک نیز Correlation Matrix ویژگی‌ها را محاسبه کردیم که به صورت شکل ۱۵ بوده است. در این ماتریس heatmap سطر و ستون‌های n 174 و n 197 خالی یا همان NaN هستند چراکه فرمول محاسبه‌ی همبستگی به صورت



زیر است و این پارامترها مقادیر تماماً یکسان در مجموعه داده داشته‌اند.

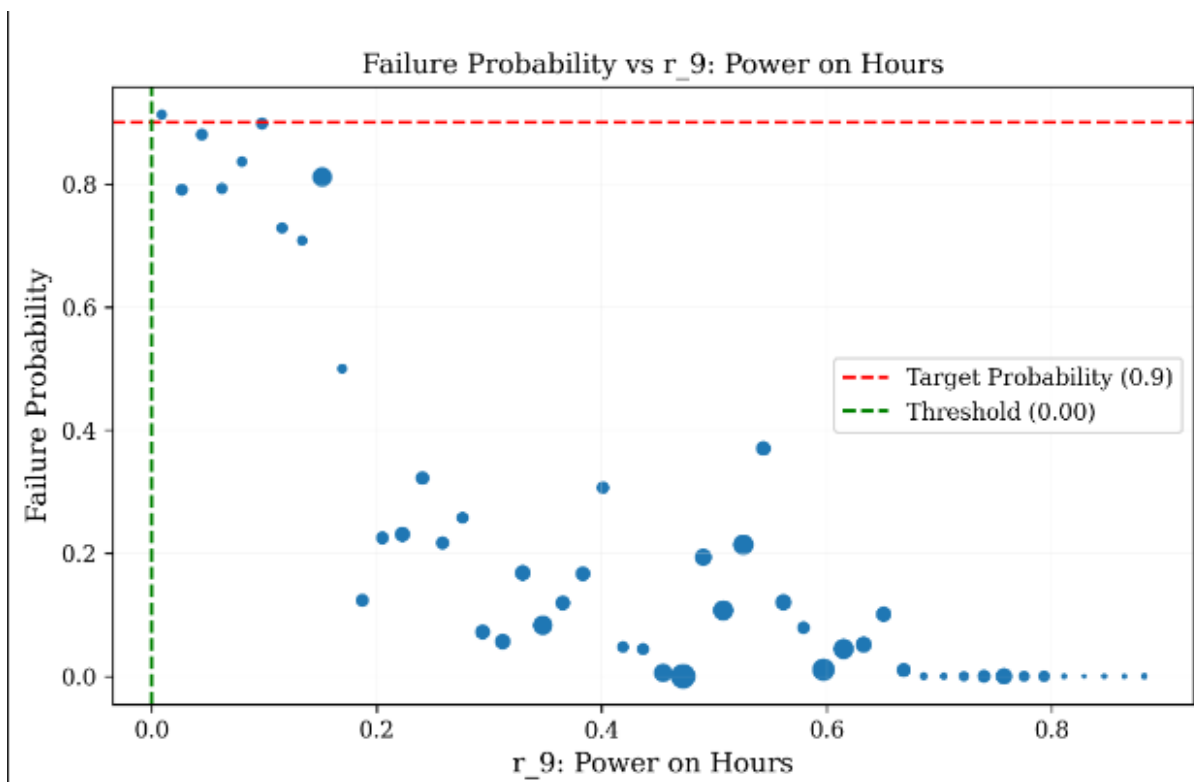
$$\text{Correlation} = \frac{\text{Cov}(x, y)}{\sigma_x \cdot \sigma_y}$$

بنابراین خواسته‌ی این بخش که شامل حد آستانه برای هر یک از ویژگی‌ها در رابطه با خرابی دیسک و میزان همبستگی و ارتباط هر یک از این پارامترهای گزارش شده با خرابی دیسک بوده است با توجه به توضیحات گفته شده به روش‌های مذکور بدست آمدند.

۳-۴-۰ بخش سوم: تشخیص ناهنجاری‌ها

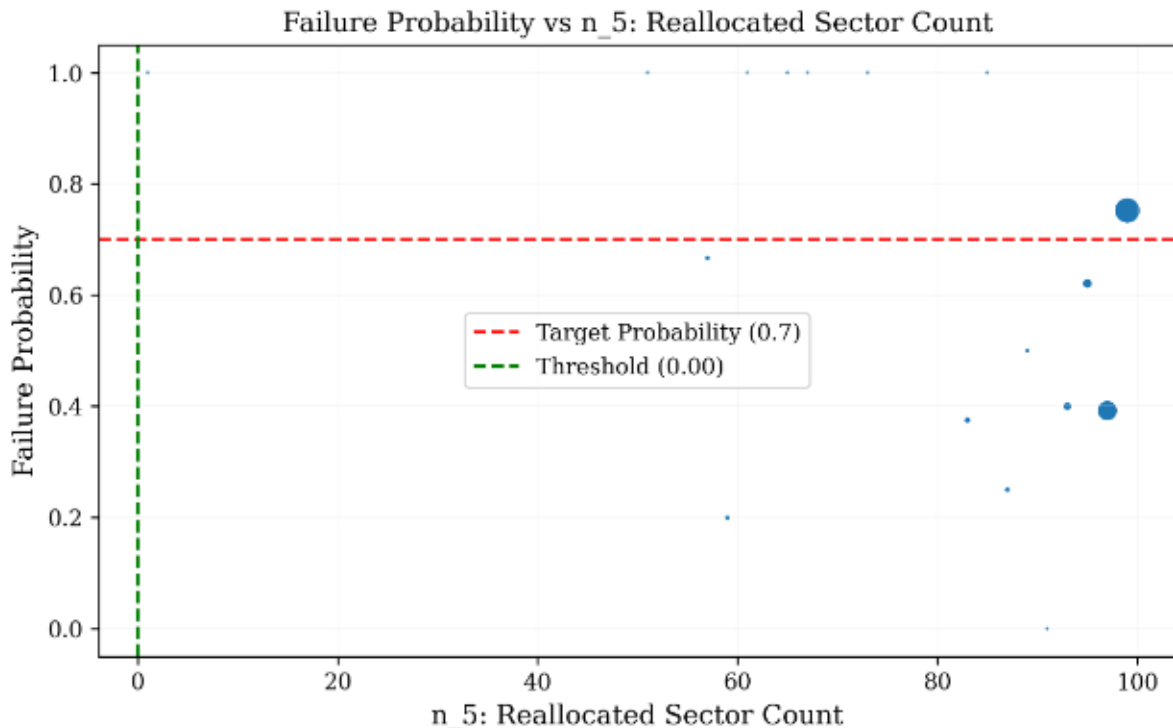
برای هر ستون از داده‌ها با توجه به دامنه هر یک از اعدادی که هر ستون می‌تواند بگیرد دامنه اعداد را به ۵۰ قسمت تقسیم می‌کنیم که این عدد قابل تغییر است و سپس در هر دسته از این اعداد احتمال اینکه هر کدام از اعداد این دسته موجب خرابی بشود را محاسبه می‌کنیم که روش محاسبه هم بدین صورت است که تعداد درایو‌های خرابی که در آن دسته قرار می‌گیرند را بر کل درایو‌های آن دسته تقسیم می‌کنیم.

که نمودار یکی از ستون‌ها را در شکل زیر مشاهده می‌کنید



شکل ۱۱: نمودار تشخیص ناهنجاری ها با توجه به احتمال برای ستون مدت زمان روشن بودن

اگر به نمودار نگاه کنید زمانی که \bullet باشد مدت زمان روشن بودن احتمال دارد که ۹۰ درصد دچار خرابی شویم که نتیجه قابل پذیرش است با توجه به نرخ خرابی بالا در ابتدای کار و همانطور که مشاهده میشود با افزایش سن این مقدار کاهش میابد



شکل ۱۲: نمودار تشخیص ناهنجاری ها با توجه به احتمال برای ستون تعداد سکتور های جابه جا شده

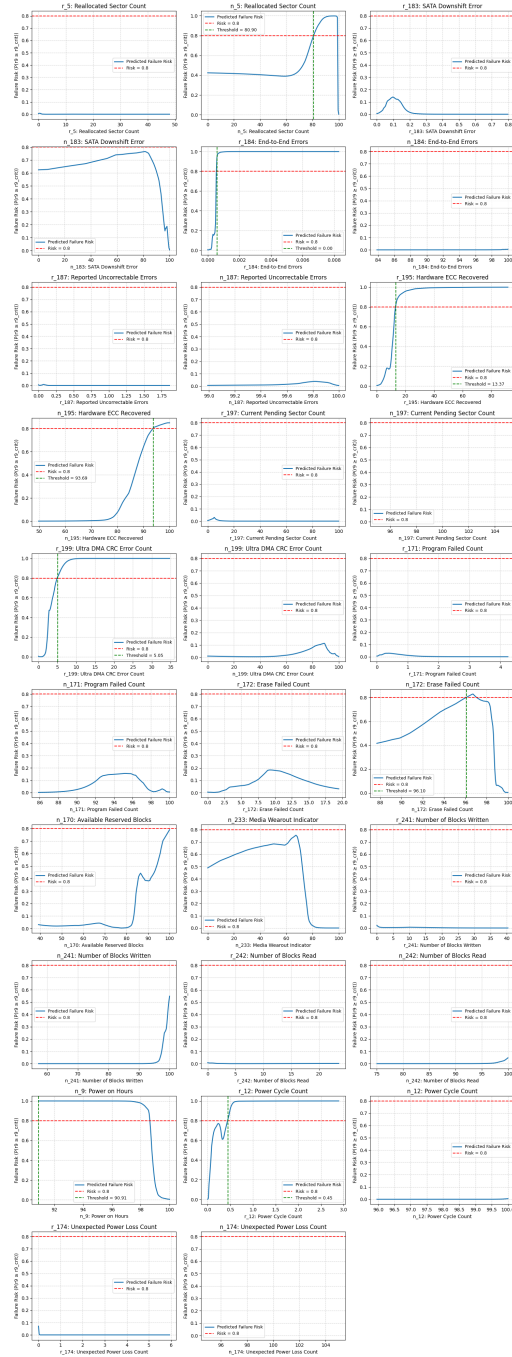
در نمودار بالا حالت اینکه احتمال ۷۰ درصد را باشد بررسی کرده ایم همانطور که مشاهده میشود مقدار حد آستانه صفر به دست آمده است که اگر دقیق به نمودار نگاه کنیم در ابتدای نمودار تعدادی از درایو ها وجود دارند که با تعداد سکتور های جابه جا شده صفر دچار خرابی شده اند به علت خرابی های زیاد که در اول کار موجود است برای حل این موضوع اگر خرابی های اولیه را در نظر نگیریم و حد آستانه را گزارش کنیم نتایج بسیار قابل قبول تر میباشند که در انتهای پروژه این کار انجام میشود.

در ادامه برای محاسبه مقادیر آستانه از مدل شبکه عصبی محاسبه شده در قسمت ۱ استفاده میکنیم و برای محاسبه نقطه بحرانی یک ویژگی کاری که انجام میدهیم آن است که بقیه ویژگی ها را برابر مقدار میانگینشان قرار می دهیم سپس ویژگی مورد بحث را برای تمام مقادیری که میتواند بگیرد مقدار آن را تغییر میدهیم و توزیع احتمال آن را به دست میاوریم و بررسی میکنیم که در چه مقداری این مقادیر از حد مورد نظر ما بیشتر هستند و از آن جایی که مدل ما مخلوطی از چند مدل است آن احتمال ها بر اساس فرمول های زیر محاسبه میشوند

$$\text{TailProb}_i = 1 - \Phi \left(\frac{r_{\text{crit}}^9 - \mu_i}{\sigma_i} \right)$$

$$\text{Risk} = \sum_{i=1}^K \pi_i \times \text{TailProb}_i$$

در نموداری زیر همه ی نمودارها را مشاهده می کنید



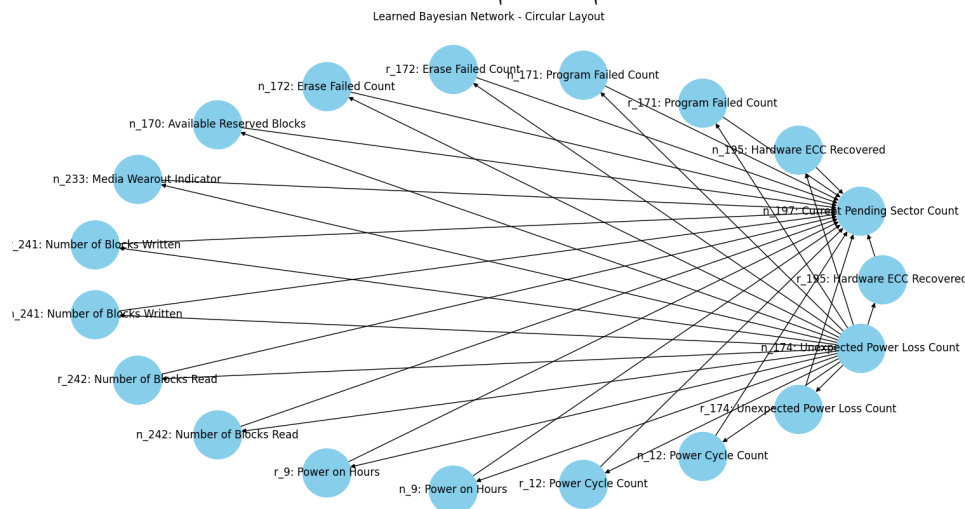
۴-۴-۰ بخش چهارم: استخراج همبستگی میان ویژگی های اسمارت

برای به دست آوردن وابستگی شرطی میان ویژگی های اسمارت از روش PGM استفاده کردیم و نکته ایی که در بخش های قبل هم مشاهده کردیم آن بود که الزاما همه ویژگی ها اطلاعات به درد

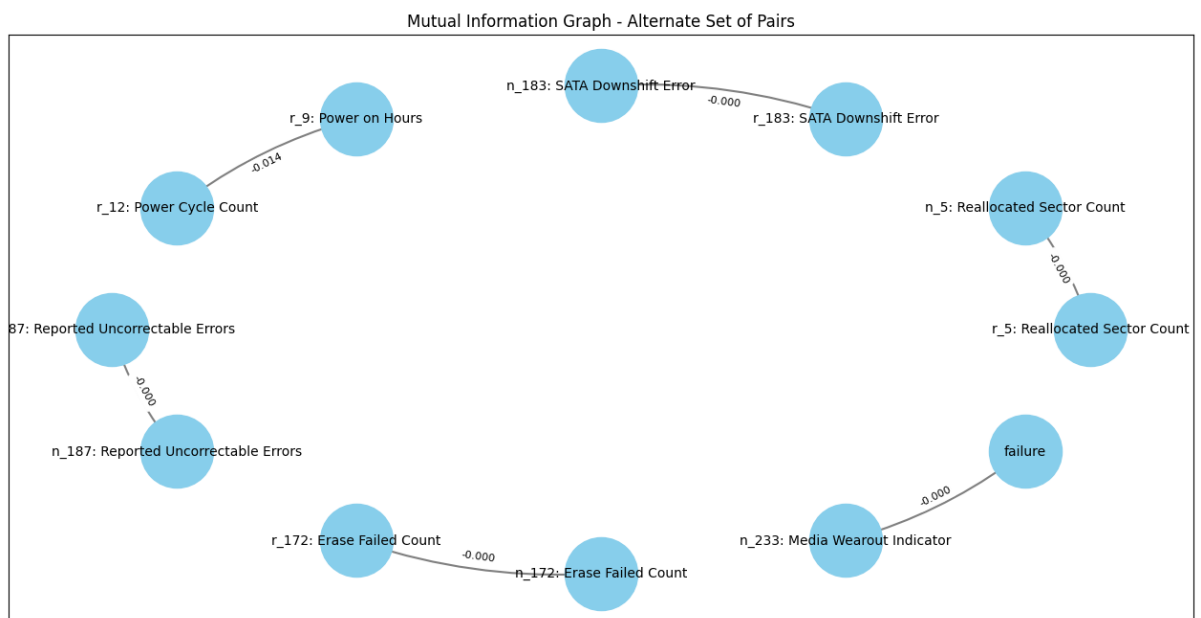
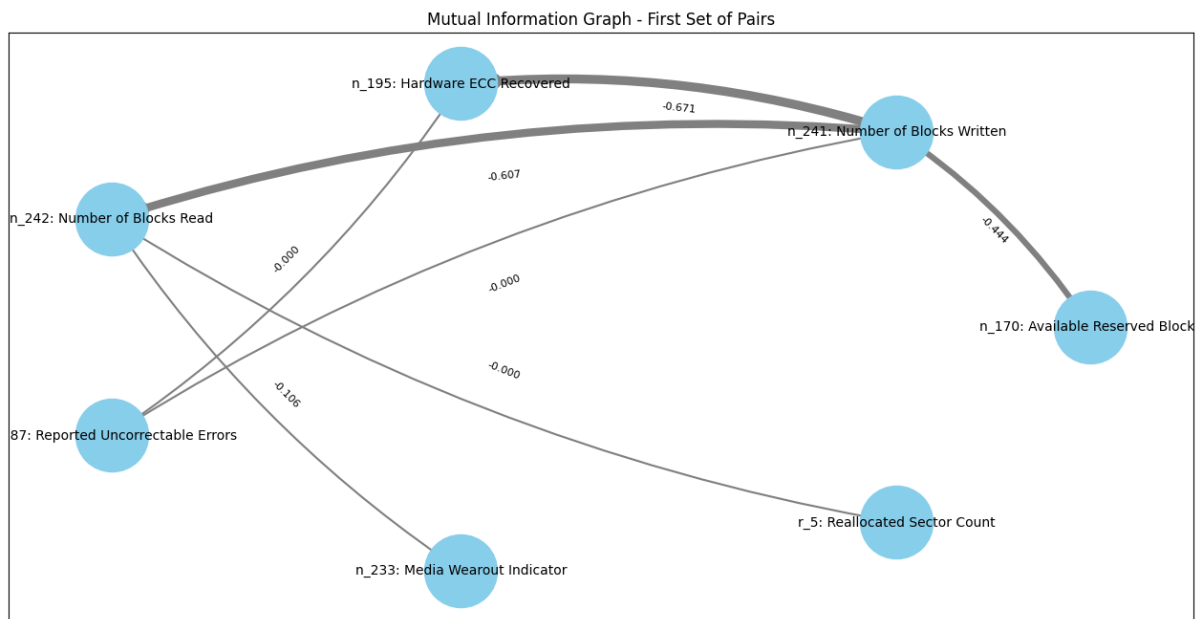
بخوری داخل خود ندارند. روند کار را اگر توضیح دهیم شبکه بیزین و اینکه محاسبه اطلاعات مشترک بین فیچرها باید داده‌ها را گسسته کنیم برای گرفتن نتیجه بهتر که داده‌ها را به بین‌های چند تایی تقسیم می‌کنیم و به هر داده داخل هر بین یک عدد نسبت می‌دهیم و اینگونه داده‌ها را گسسته می‌کنیم سپس سعی می‌کنیم با استفاده از الگوریتم تپه نوردی از یک شبکه خالی شروع به حرکت کنیم و اضافه کردن یا پاک کردن یال سعی می‌کند امتیاز مدل را افزایش دهیم که امتیاز مدل با استفاده از امتیاز BIC محاسبه می‌شود که

$$\text{BIC} = \ln(L) - \frac{d}{2} \ln(n)$$

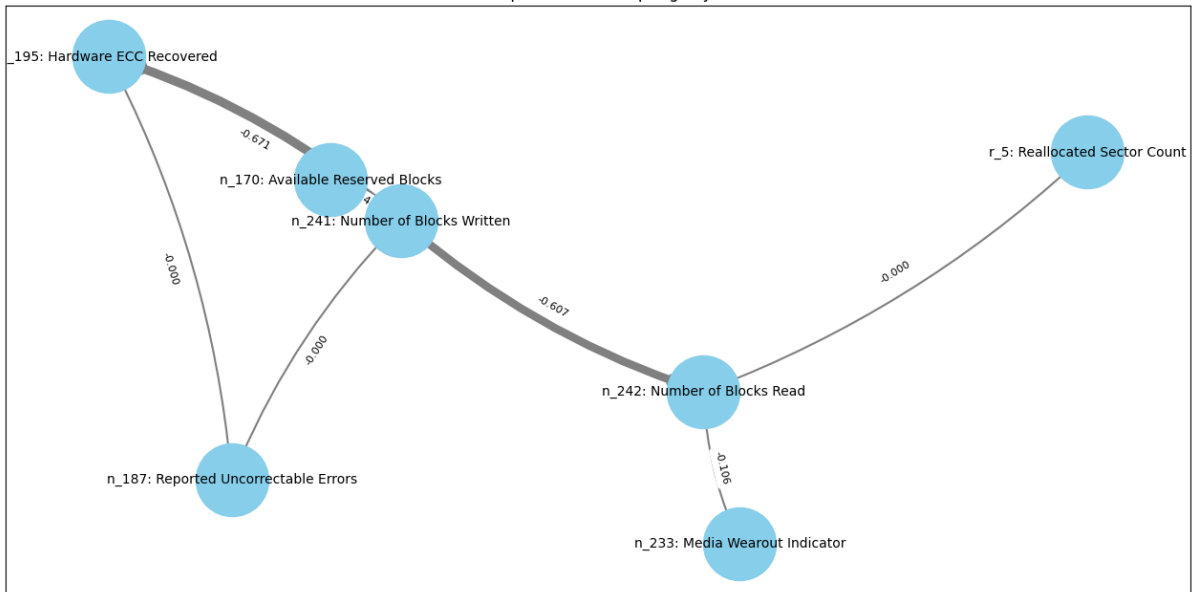
پس از محاسبه بهترین گراف را رسم می‌کنیم که در زیر شکل‌های آن را مشاهده می‌کنید



برای محاسبه وابستگی بین دو متغیر تصادفی از فرمول اطلاعات متقابل استفاده می‌کنیم و نمودار اطلاعات متقابل بین جفت ویژگی‌ها را رسم می‌کنیم که به انتخاب دو نمونه نمودار در زیر آورده شده است



MI Graph (First Set) - Spring Layout



۵-۰ نتیجه گیری

در مورد نتایج بخش اول متوجه میشویم که روش رگرشن نتایج خوبی برای ما به همراه ندارد البته که با توجه به مجموعه داده در دسترس توانستیم دقت ۹۰ درصد بر روی پیش بینی ها بگیریم اما همچنان نتایج راضی کننده نیست. که در ادامه قصد داریم از شبکه عصبی استفاده کنیم تا بتوانیم نتایج خود را بهبود دهیم. در مورد قسمت سوم موردی که مشاهده میشود که قبلا هم ذکر شده بود عوامل خرابی های اولیه یا همان Infant Mortality باعث میشوند که گزارش های مربوط به این بخش نتایج مورد انتظار ما را نداشته باشد هر چند که نتایج هم تا به اینجا قابل قبول هستند پس در ادامه قصد داریم که نتایج به دست آمده را یکبار هم با حذف خرابی های اولیه گزارش نماییم. در مورد قسمت چهارم نیز با توجه به روش های بیزین و بررسی روش های موجود در انتهای پروژه وابستگی های شرطی را میابیم.

Yu Cai; Saugata Ghose; Erich F. Haratsch; Yixin Luo; Onur Mutlu, “*Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives*” in *IEEE* doi: [10.1109/JPROC.2017.2713127](https://doi.org/10.1109/JPROC.2017.2713127).

Jacob Alter, Ji Xue, Alma Dimnaku, Evgenia Smirni, “*SSD failures in the field: symptoms, causes, and prediction models*” in *ACM* doi: [10.1145/3295500.3356172](https://doi.org/10.1145/3295500.3356172).

Chandranil Chakrabortii, Heiner Litz, “*Improving the accuracy, adaptability, and interpretability of SSD failure prediction models*” in *ACM*, doi.org/10.1145/3419111.3421300.

Yuqi Zhang, Wenwen Hao, Ben Niu, Kangkang Liu, Shuyang Wang, Na Liu, Xing He, Yongwong Gwon, Chankyu Koh, “*Multi-view feature-based SSD failure prediction: what, when, and why*” in *ACM*, [link](#).