

به نام خدا



دانشکده مهندسی مکانیک

# پایش سلامت پمپ‌های گریز از مرکز با استفاده از شبکه‌های عصبی

پروژه تخصصی کارشناسی  
رشته مهندسی مکانیک

مهدی عبدالله چالکی

استاد راهنما:

دکتر علی صدیقی

شهریور ماه ۱۴۰۰

تقدیم به:

خانواده‌ی عزیزم که همواره من را در این مسیر حمایت کردند و مشتاقانه آرزومند موفقیت‌م بودند.

## تشکر و قدردانی:

با تشکر ویژه از استاد گرانقدر جناب دکتر علی صدیقی بابت راهنمایی‌ها و حمایت‌های دلسوزانه در تمامی مراحل انجام این پروژه و آقایان مهدی رباطی، دانیال معروفی معروف آبادی، محمد جامی مقدم و علیرضا مصطفوی بابت تمام راهنمایی‌هایی که در این پروژه به من کردند.

## چکیده

در این پروژه یک سامانه‌ی تشخیص عیوب پمپ‌های سانتریفیوژی طراحی و ساخته شده است. این سامانه متشکل از یک بخش داده برداری و یک بخش پردازش مرکزی است که به صورت بی‌سیم با یکدیگر ارتباط دارند. بخش داده برداری شامل شتاب‌سنج، میکروکنترلر و ماژول فرستنده‌ی بی‌سیم و باتری است و بخش پردازش مرکزی نیز از یک ماژول گیرنده‌ی بی‌سیم و یک کامپیوتر رزبری پای تشکیل شده است.

برای تشخیص عیب پمپ‌های دوار، از شبکه‌های عصبی پیچشی<sup>۱</sup> استفاده شده است. سپس دستگاهی کوچک برای داده‌برداری طراحی و ساخته شده و شبکه‌ی عصبی به وسیله‌ی این داده‌ها آموزش داده شده است. همچنین برای اضافه کردن امکان تشخیص عیوب جدید به این سامانه پس از راه‌اندازی، سناریویی طراحی گردیده و شبکه‌ی عصبی با الگوریتم‌های خوشه‌بندی ترکیب شده‌اند.

در نهایت نیز با راه‌اندازی بخش مربوط به پردازش مرکزی، زنجیره‌ی شامل تجهیز صنعتی، دستگاه داده‌برداری و نیز بخش پردازش مرکزی کامل شده و فرایند برخط تشخیص عیب و نمایش اطلاعات به کاربر صورت می‌گیرد.

**واژه‌های کلیدی:** شبکه‌های عصبی، پایش وضعیت، تشخیص عیب، ماشین‌های دوار، یادگیری برخط

---

<sup>1</sup> Convolutional Neural Networks

## فهرست مطالب

فصل اول	۱
۱-۱- مقدمه	۲
۲-۱- تعریف مساله	۲
۳-۱- هدف از طرح مورد نظر، ضرورت انجام آن	۳
۴-۱- روش‌های اجرایی انجام پروژه:	۴
۵-۱- روش‌های پیشین	۴
۶-۱- مقایسه با پروژه‌ی پیشین	۶
فصل دوم	۸
۱-۲- مقدمه	۹
۲-۲- شبکه‌های عصبی	۹
۱-۲-۲- شبکه‌ی عصبی	۹
۲-۲-۲- شبکه‌ی عصبی پیچیده	۱۰
۳-۲- تهیه‌ی مجموعه‌ی داده‌ها	۱۲
۴-۲- مدل به کار گرفته شده در این پروژه	۱۶
۵-۲- فرایند آموزش شبکه‌ی عصبی	۱۷
۱-۵-۲- کتابخانه‌های مورد استفاده	۱۷
۲-۵-۲- آموزش شبکه‌ی عصبی	۱۸
۳-۵-۲- روش K-fold	۱۹
۶-۲- به‌روزرسانی شبکه‌ی عصبی و تشخیص عیوب جدید	۲۰
۱-۶-۲- روش به کار گرفته شده	۲۰
۲-۶-۲- الگوریتم DBSCAN	۲۵
۳-۶-۲- الگوریتم K-Means	۲۶
۷-۲- سخت افزار بخش داده‌برداری	۲۸
۲-۷-۲- باتری	۲۹

۳۰.....	۳-۷-۲- مازول ارتباطی.....
۳۲.....	۴-۷-۲- میکرو کنترلر .....
۳۵.....	۵-۷-۲- طراحی محفظه‌ی دستگاه.....
۳۷.....	۸-۲- سخت افزار بخش پردازش مرکزی.....
۳۷.....	۱-۸-۲- توضیحات کلی.....
۳۸.....	۲-۸-۲- انتخاب کامپیوتر.....
۳۹.....	۹-۲- نرم‌افزار بخش پردازش مرکزی.....
۴۱.....	فصل سوم.....
۴۲.....	۱-۳- نتایج فرایند K-fold.....
۴۲.....	۲-۳- ماتریس درهم‌ریختگی.....
۴۳.....	۳-۳- بصری سازی خروجی لایه‌ها.....
۵۱.....	مراجع.....

## فهرست اشکال

### فصل ۱

شکل ۱-۱: دستگاه طراحی شده در پروژه‌ی دستگاه هوشمند تشخیص عیوب در ماشین‌های دوار ..... ۷

### فصل ۲

شکل ۲-۱: تصویر شماتیک یه شبکه‌ی عصبی ساده ..... ۱۰

شکل ۲-۲: تصویر شماتیک یک شبکه‌ی عصبی پیچیده مورد استفاده در تشخیص تصاویر ..... ۱۱

شکل ۲-۳: بستر آزمایشگاهی حلقه-بسته مورد استفاده در پروژه ..... ۱۳

شکل ۲-۴: (a) عیب پدیدگی نوک پره در اندازه‌ی کوچک - (b) عیب پدیدگی نوک پره در اندازه‌ی بزرگ - (c) شبیه‌سازی ترک در پره ..... ۱۵

شکل ۲-۵: شماتیک معماری شبکه‌ی طراحی شده‌ی نهایی مورد استفاده در پروژه ..... ۱۷

شکل ۲-۶: فرایند اعتبار سنجی مدل شبکه‌ی عصبی طراحی شده توسط الگوریتم K-fold ..... ۱۹

شکل ۲-۷: الگوریتم طی‌شده برای افزودن عیب جدید به برجسب‌های قبلی ..... ۲۲

شکل ۲-۸: ماتریس درهم‌ریختگی با پنج کلاس ..... ۲۳

شکل ۲-۹: بصری‌سازی خروجی لایه‌ی آخر در حالت شبکه‌ی آموزش داده شده با پنج کلاس ..... ۲۳

شکل ۲-۱۰: خروجی پیش‌بینی شده برای عیب جدید - تست شده بر روی ۱۰۰ داده ..... ۲۴

شکل ۲-۱۱: بصری‌سازی خروجی کاهش‌یافته‌ی لایه‌ی FC2 توسط روش t-SNE ..... ۲۵

شکل ۲-۱۲: خوشه بندی داده‌ها توسط الگوریتم DBSCAN ..... ۲۶

شکل ۲-۱۳: نمودار جمع مربعات خطا برای دسته‌بندی با تعداد خوشه‌های مختلف توسط الگوریتم K-Means ..... ۲۷

شکل ۲-۱۴: ماژول شتاب‌سنج CJMCU LIS3DSH ..... ۲۹

شکل ۲-۱۵: ماژول ارتباطی XBee ..... ۳۱

شکل ۲-۱۶: آنتن ارتباط بی‌سیم با فرکانس ۲.۴ گیگاهرتز و بهره‌ی 3dBi ..... ۳۲

شکل ۲-۱۷: تبدیل U.FL به RPSMA ..... ۳۲

شکل ۲-۱۸: میکروکنترلر رزبری پای پیکو ..... ۳۳

شکل ۲-۱۹: شماتیک ارتباط میان اجزای بخش قابل حمل ..... ۳۴

شکل ۲-۲۰: شماتیک اتصالات الکترونیکی قطعات اجزای بخش قابل حمل ..... ۳۵

شکل ۲-۲۱: نقشه‌ی انفجاری بدنه‌ی دستگاه ..... ۳۶

شکل ۲-۲۲: مدل نهایی پرینت شده‌ی بدنه‌ی دستگاه ..... ۳۷

شکل ۲-۲۳: تصویر کامپیوتر Raspberry Pi 4 ..... ۳۸

### فصل ۳

شکل ۳-۱: ماتریس در هم ریختگی داده‌های تست ..... ۴۳

شکل ۳-۲: بصری‌سازی فرایند جداسازی ۵۰۰ داده‌ی متعلق به دسته‌های متفاوت در لایه‌های شبکه‌ی عصبی ..... ۵۰

## فهرست جداول

### فصل ۱

جدول ۱-۱: ماتریس درهم ریختگی برای روش مبتنی بر فاصله ماهالانوبیس..... ۶

جدول ۱-۲: امتیاز معیارهای عملکرد برای روش مبتنی بر فاصله ماهالانوبیس..... ۶

### فصل ۲

جدول ۲-۱: مجموعه‌ی داده‌های مورد استفاده در فرایند آموزش و تست شبکه‌ی عصبی..... ۱۴

جدول ۲-۲: مشخصات فنی ماژول XBee [22]..... ۳۱

جدول ۲-۳: مشخصات میکروکنترلر RP2040 [23]..... ۳۳

جدول ۲-۴: مشخصات کامپیوتر Raspberry Pi 4 [26]..... ۳۹

### فصل ۳

جدول ۳-۱: نتایج اعتبارسنجی مدل توسط فرایند K-fold..... ۴۲



## فصل اول

### کلیات

## ۱-۱- مقدمه

## تعمیر و نگهداری پیش‌بینانه (Predictive Maintenance)

تعمیر و نگهداری پیش‌بینانه روشی است که از ابزارها و تکنیک‌های تجزیه و تحلیل داده‌ها برای تشخیص ناهنجاری‌ها در عملکرد و نقص‌های احتمالی در تجهیزات و فرآیندها استفاده می‌کند تا بتوان پیش از خرابی تجهیزات، آن‌ها را تعمیر کرد. تمامی دستگاه‌ها از هر شکلی مستعد آسیب دیدن و خراب شدن هستند و عمر آن‌ها با توجه به فرکانس نگهداری‌شان افزایش یا کاهش می‌یابد. تعمیر و نگهداری پیش‌بینانه سعی دارد تا جلوی تعمیرات واکنشی (reactive – تعمیراتی که پس از خرابی دستگاه بر روی آن انجام می‌گیرد تا مجدداً قابل استفاده شود) را بگیرد و همزمان فرکانس تعمیرات را به گونه‌ای تنظیم کند تا هزینه‌ی زیادی برای تعمیرات پیش‌بینانه به هدر نرود.

یک مطالعه‌ی مشترک صورت‌گرفته توسط وال استریت ژورنال و امرسون نشان داده است که زمان‌های از کار افتادگی برنامه ریزی نشده سالانه ۵۰ میلیارد دلار برای تولیدکنندگان صنعتی هزینه دارد. خرابی تجهیزات، عامل ۴۲ درصد از این از کارافتادگی‌ها است.

پژوهشگران در این زمینه‌ی تحقیقاتی به دنبال توسعه ابزارها، دستگاه‌ها و الگوریتم‌هایی هستند که بتوانند عیوب ادوات دوار همچون پمپ، کمپرسور و موتور الکتریکی را پیش از آن‌که منجر به خسارت‌های زیاد شود، تشخیص دهند.

به طور متوسط نگهداری و تعمیرات پیشگیرانه از کارافتادگی ناگهانی دستگاه‌ها را تا ۷۰٪ کاهش می‌دهد. بهره‌وری را تا ۲۵٪ افزایش داده و تا ۲۵٪ هزینه‌های بنگاه اقتصادی را کاهش می‌دهد. اندازه‌ی بازار این فناوری با گسترش فناوری‌هایی همچون اینترنت اشیا به شدت رو به گسترش است. طبق گزارش موسسه Market Research Future، ارزش این بازار در سال ۲۰۲۲ به رقمی بالغ بر ۶.۳ میلیارد دلار خواهد رسید.[1]

## ۱-۲- تعریف مساله

امروزه ماشین‌های دوار تقریباً در همه صنایع استفاده می‌شوند. این ماشین‌ها همانند سایر ماشین‌آلات صنعتی، پس از مدتی کار کردن نیاز به تعمیر و تعویض قطعات دارند. هر تعمیر سبک و یا سنگین، باعث افت بهره‌وری شده و نرخ رشد را کاهش می‌دهد. خرابی ناگهانی ماشین‌های دوار و نیاز آن‌ها به تعمیر منجر به توقف تولید، آسیب احتمالی به بخش‌های مختلف ماشین‌آلات و افزایش ضایعات تولید می‌شود.[2]

با ورود به نسل چهارم انقلاب صنعتی که یکی از حوزه‌های آن اینترنت اشیاء صنعتی است، تمرکز بیشتری بر روی بهره‌وری، هزینه‌های عملیاتی و ... وجود دارد. در اینترنت اشیاء صنعتی، به دنبال آن هستیم تا با استفاده از سنسورهای مناسب بتوانیم داده‌های ارزشمندی از یک ماشین جمع‌آوری کرده و با استفاده از روش‌های تحلیلی پیشرفته، به دید تازه‌ای از سیستم برسیم. پیش‌بینی می‌شود که اینترنت اشیاء صنعتی بتواند بهره‌وری تولیدکنندگان را تا ۳۰ درصد افزایش دهد که مقدار قابل توجهی است.[3]

طبق نظرسنجی انجمن اقتصاد جهانی، امروزه بیشترین کاربرد اینترنت اشیاء صنعتی در حوزه نگهداری پیش‌بینانه انجام می‌گیرد. با تشخیص زودهنگام مشکل پیش از آنکه دستگاه از کار بیفتد، صاحبان صنایع می‌توانند هزینه‌های نگهداری را کم کرده، عمر ماشین آلات را افزایش داده و زمان تعمیرات را به حداقل برسانند.

در این پروژه، به دنبال آن هستیم که بتوانیم با استفاده از سیستم‌های نوین و به‌روز، مشکل پیش‌بینی و پایش سلامت ماشین‌آلات دوار صنعتی را حل کرده و هزینه‌های ناشی از آن را کاهش دهیم.

### ۱-۳- هدف از طرح مورد نظر، ضرورت انجام آن

موسسه جهانی مک‌کینزی<sup>۲</sup> در گزارشی پیش‌بینی کرده است که صرفه‌جویی حاصل از نگهداری و تعمیر پیش‌بینانه تا سال ۲۰۲۵، به عددی بین ۲۴۰ تا ۶۳۰ میلیارد دلار خواهد رسید.[4] این مبلغ چشم‌گیر، توجه محیط آکادمیک را به خود جلب کرده و این حوزه را به مبحثی بسیار داغ تبدیل کرده است.

با توجه به مشکلاتی که در بخش قبل مطرح شد، پیش‌بینی و تشخیص زودهنگام عیوب ماشین‌های دوار یکی از نیازهای اساسی صنعت امروز به شمار می‌رود. با استفاده از نگهداری و تعمیر پیش‌بینانه، می‌توان جلوی بسیاری از هزینه‌های ناشی از خراب شدن ناگهانی دستگاه‌ها و توقف تولید را گرفت. در این صورت، می‌توان تدابیر لازم برای تعمیر و جایگزینی ماشین‌آلات در خط اندیشید و زمان لازم برای تعمیرات را با تشخیص به موقع به حداقل برساند. همچنین اگر بتوان هر چه سریع‌تر ایرادات قطعه معیوب پیدا کرد، احتمال آسیب رسیدن به دیگر قطعات دستگاه نیز کاهش چشم‌گیری پیدا خواهد کرد و نیز هزینه تعمیر دستگاه بسیار کمتر خواهد بود.

هدف از انجام این پروژه، پایش سلامت برخط و عیب‌یابی این ماشین‌آلات پیش از وقوع آسیب جدی با استفاده از سیستم اندازه‌گیری شامل حسگرها، مدارهای بهسازی سیگنال و ... و سپس انتقال برخط داده‌ها برای بررسی وضعیت سلامتی ماشین مورد نظر است تا در صورت وجود هرگونه عیب و پیش از کار افتادن سیستم، هشدارهای لازم را بدهیم. این مهم با پردازش داده‌های دریافت شده از سیستم و یک شبکه عصبی صورت می‌گیرد و تنها بر داده متکی است.

<sup>2</sup> McKinsey Global Institute

## ۴-۱- روش‌های اجرایی انجام پروژه:

امروزه با توجه به روند کاهشی قیمت ریزپردازشگرها و حسگرها و نیز گسترش زیرساخت‌های فناوری اطلاعات، قابلیت اتصال تجهیزات صنعتی به شبکه و ارسال داده‌های مرتبط بیش از پیش مهیا شده است. حجم گسترده داده‌ها این امکان را فراهم می‌سازد تا به جای مدل‌سازی تجهیزات صنعتی بر اساس قوانین فیزیکی، مدل‌سازی بر اساس داده‌ها انجام پذیرد. در نهایت مدل‌های به دست آمده را می‌توان برای تشخیص عیوب مورد استفاده قرار داد. [5]

تجزیه و تحلیل سیستم توسط انسان که فرایندی وقت‌گیر و غیرقابل اعتماد است، به مرور در حال جایگزین شدن با روش‌های هوشمند عیب‌یابی است. با پیشرفت علم هوش مصنوعی و افزایش توان پردازشی سیستم‌های کامپیوتری، استفاده از روش‌های یادگیری ماشین نیز در حوزه‌های صنعتی و به خصوص برای تشخیص عیب افزایش یافته است. مدل‌های یادگیری عمیق با توانایی نگاشت غیر خطی چندلایه خود، توانسته اند دقت عیب‌یابی هوشمند را افزایش دهند. [6]

در این پروژه، از شبکه‌های عصبی عمیق برای پایش سلامت ماشین‌های دوار استفاده می‌شود. مزیت این روش، داده محور بودن آن است؛ بدان معنی که مانند روش‌های سنتی، به دانش پیشین در مورد سیستم مورد نظر و نظر فرد متخصص نیازی نداریم.

دیگر ویژگی مهم سیستم طراحی شده در این پروژه، برخط بودن آن است. دریافت و پردازش داده‌ها به صورت برخط صورت گرفته و با کمک شبکه عصبی آموزش داده شده، سیستم در هر لحظه پایش شده و به محض ظهور عیب، هشدار داده خواهد شد.

در گام اول این پروژه، سخت افزار لازم برای داده برداری و انتقال داده‌ها به پایگاه داده فراهم می‌گردد. سپس داده برداری از تجهیز هدف در شرایط مختلف کاری انجام شده و این داده‌ها برای آموزش شبکه عصبی مورد استفاده قرار می‌گیرد. در نهایت، شبکه عصبی آموزش داده شده به صورت برخط اجرا شده و در شرایط کاری مختلف صحت سنجی خواهد شد و اصلاحات لازم صورت خواهد پذیرفت.

## ۵-۱- روش‌های پیشین

از ابتدای شروع تحقیقات در این حوزه، ابتدا محققین سعی داشتند تا با استفاده از مدل‌سازی ریاضی سیستم‌های دوار و قطعات آن‌ها مانند چرخ‌دنده‌ها و بلبرینگ‌ها، و نیز بهره‌گیری از تکنیک‌های پردازش سیگنال (مواردی چون تبدیل فوریه، تحلیل Cepstrum, Spectrum و ... ) به تشخیص عیوب بپردازند. [7][8] این نوع از پایش سیستم‌ها علاوه بر نیاز به افراد متخصص، نیاز به دانش قبلی در مورد سیستم مورد مطالعه دارد. برای مثال تعداد

سازمان‌های درون بلبرینگ و سائز آن‌ها و یا تعداد دنده‌های یک چرخدنده، از پارامترهای تاثیر گذار در مدل‌سازی هستند که معمولاً اطلاع یافتن از آن‌ها در محیط صنعتی و برای هر ماشین، کاری دشوار است؛ اما اطلاع از آن‌ها برای پردازش سیگنال و تشخیص عیب احتمالی، امری گریز ناپذیر است. این نقاط ضعف در کنار امکان ناپذیر بودن مدل‌سازی سیستم‌های پیچیده، باعث شد تا روش‌های مبتنی بر داده<sup>۳</sup> مورد توجه دانشمندان قرار گیرند.

در اوایل دهه ۲۰۰۰ میلادی، روش‌های مبتنی بر علم آمار (مانند رگرسیون و الگوریتم‌های خوشه‌بندی) مورد مطالعه محققین برای استفاده در حوزه تعمیر و نگهداری پیش‌بینانه قرار گرفتند. مزیت نسبی این الگوریتم‌ها نسبت به روش‌های پیشین، مبتنی بودن بر داده‌ها است که باعث می‌شود نیازی به مدل‌سازی ریاضی و تکنیک‌های پردازش سیگنال نباشد. از این روش‌های آماری، بیشتر به منظور تشخیص وجود عیب<sup>۴</sup> و نه تشخیص نوع عیب<sup>۵</sup> استفاده می‌شد.

به مرور زمان و با افزایش قدرت و کاهش قیمت پردازنده‌ها، روش‌های هوش مصنوعی که سال‌ها قبل الگوریتم‌های آن‌ها توسعه داده شده بود، مجدداً مورد توجه قرار گرفتند. در حوزه تعمیر و نگهداری پیش‌بینانه، محققین با استفاده از الگوریتم‌های هوش مصنوعی از جمله ماشین‌های بردار پشتیبان<sup>۶</sup>، درخت‌های تصمیم‌گیری<sup>۷</sup>، اتوانکودرها<sup>۸</sup> و شبکه‌های عصبی<sup>۹</sup> توانستند به پیشرفت‌های چشمگیری دست پیدا کنند و با دقت‌هایی به مراتب بالاتر، به تشخیص نوع عیب پردازند. [9][10]

در دهه‌ی ۲۰۱۰ میلادی، الگوریتم‌های یادگیری عمیق<sup>۱۰</sup> توسعه‌ی بیشتری یافتند و روش‌های مبتنی بر آن‌ها مانند شبکه‌های عصبی پیچیده<sup>۱۱</sup> و اتوانکودرهای انباشته<sup>۱۲</sup> مورد توجه قرار گرفتند. کارکرد اصلی این شبکه‌ها، معمولاً برای تشخیص اجسام و دسته‌بندی تصاویر دو بعدی است. اما محققین توانسته‌اند در حالت یک بعدی نیز از آن‌ها استفاده کرده و برای تشخیص صدا و نویز و دیگر مواردی که سیگنال‌های آن‌ها به صورت خطی است نیز از این شبکه‌های بهره ببرند. [11][12][13]

در پروژه‌ی انجام شده، انواع روش‌ها و مقالات به صورت مفصل مورد بررسی قرار گرفتند. با توجه به نتایج درخشان و دقت بالای الگوریتم‌های CNN، تصمیم بر آن شد تا این نوع از معماری شبکه‌های عصبی مورد استفاده قرار گیرد.

<sup>3</sup> Data-driven

<sup>4</sup> Anomaly detection

<sup>5</sup> Fault classification

<sup>6</sup> Support Vector Machines

<sup>7</sup> Decision trees

<sup>8</sup> Auto-encoders

<sup>9</sup> Neural networks

<sup>10</sup> Deep Learning

<sup>11</sup> Convolutional Neural Networks

<sup>12</sup> Stacked Auto-Encoders

از مزایای این روش می‌توان به عدم نیاز به تخصص در امر مدل‌سازی ماشین‌های دوار در نرم‌افزارهای شبیه‌سازی و همچنین عدم نیاز به فرد متخصص در محل سیستم نصب شده برای تحلیل سیگنال‌های دریافتی اشاره کرد.

### ۱-۶- مقایسه با پروژه‌ی پیشین

سال گذشته و در پروژه‌ی کارشناسی با عنوان "دستگاه هوشمند تشخیص عیوب در ماشین‌های دوار" که به همت آقایان علیرضا مصطفوی و سید مهدی حسینی معصوم در دانشکده‌ی مهندسی مکانیک دانشگاه تهران انجام گرفت، سعی شده بود تا دستگاهی طراحی و ساخته شود که پس از نصب شدن بر روی یک ماشین دوار، به طور کاملاً خودکار (یادگیری برخط) ابتدا الگوهای موجود در داده‌های مربوط به وضعیت سالم ماشین دوار را استخراج کرده و هر زمان که الگوها از حالت سالم فاصله بگیرند، اخطار وجود عیب در سیستم را صادر کند.

ایشان در آن پروژه تلاش کرده بودند تا با استخراج ۲۹ ویژگی از یک سیگنال ارتعاشی (شامل ۱۰ ویژگی در حوزه‌ی زمان، ۳ ویژگی در حوزه‌ی فرکانس و ۱۶ ویژگی در حوزه‌ی زمان-فرکانس)، و در نظر گرفتن فاصله‌ی مایلانوبیس به عنوان معیاری برای تشخیص معیوب بودن سیگنال‌ها، به این مهم دست یابند.

دقت این روش در دو جدول زیر نمایش داده شده است:

جدول ۱۱ :- ماتریس درهم ریختگی برای روش مبتنی بر فاصله مایلانوبیس

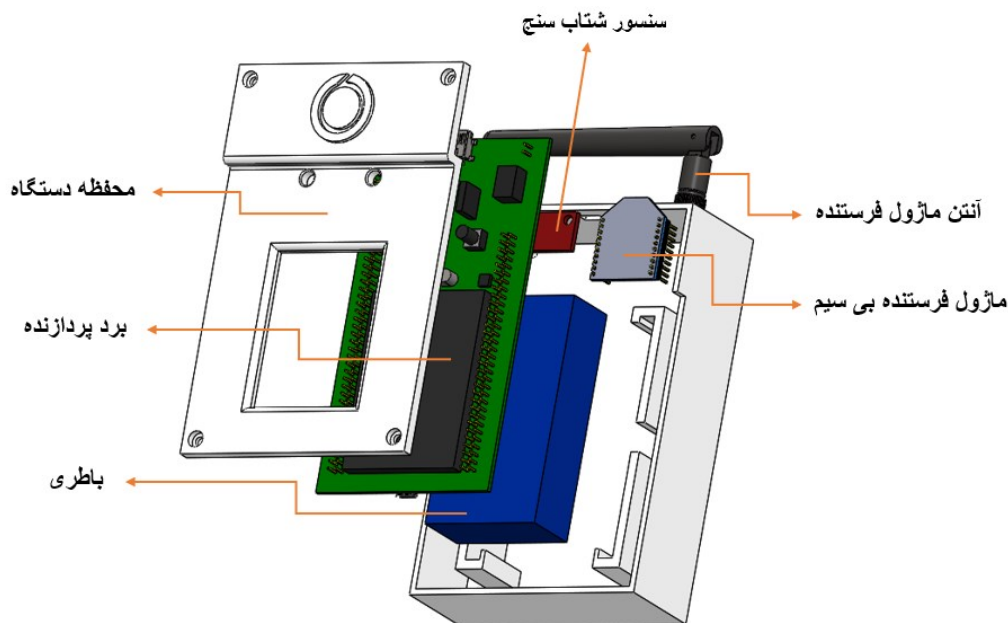
	پیش بینی سالم	پیش بینی معیوب
واقعا سالم	(منفی صحیح) 159	(مثبت غلط) 11
واقعا معیوب	(منفی غلط) 1	(مثبت صحیح) 209

جدول ۲۱ :- امتیاز معیارهای عملکرد برای روش مبتنی بر فاصله مایلانوبیس

معیار	Precision	Recall	Accuracy	F1 Score
امتیاز (از 1)	0/950	0/995	0/968	0/972

همان طور که ملاحظه می‌شود، این سیستم قادر است تا با دقت بسیار خوبی حالت‌های معیوب یک سیستم را تشخیص دهد.

همچنین این پروژه دارای یک فاز سخت‌افزاری نیز بوده‌است که شامل طراحی یک دستگاه قابل حمل و پیاده‌سازی الگوریتم‌های توسعه داده شده بر روی یک میکروکنترلر است.



شکل ۱-۱: دستگاه طراحی شده در پروژه‌ی دستگاه هوشمند تشخیص عیوب در ماشین‌های دوار

در این دستگاه از یک میکروکنترلر STM32F429zi برای پردازش اطلاعات استفاده شده است. این برد دارای یک صفحه نمایش لمسی نیز می‌باشد که اطلاعاتی شامل درصد احتمال وجود عیب را نمایش می‌دهد. این سامانه‌ی پایش سلامت که رهیافت کلی آن بر اساس ایده‌ی پردازش محلی سیگنال‌ها<sup>۱۳</sup> بوده است، دارای مزایا و معایبی نسبت به سامانه‌ی طراحی شده بر اساس پردازش مرکزی سیگنال‌ها<sup>۱۴</sup> است.

از مزایای پروژه‌ی پیشین می‌توان به کاهش هزینه‌های سخت‌افزاری و کاهش پهنای باند مورد نیاز اشاره کرد (به دلیل عدم نیاز به سرور و عدم نیاز به ارسال سیگنال‌های ارتعاشی به مرکز پردازش) همچنین از معایب این سیستم می‌توان به قدرت پردازشی محدود میکروکنترلر، محدودیت در ذخیره‌ی اطلاعات و نیز عدم امکان تعیین نوع عیب یاد کرد.

از مزیت‌های پروژه‌ی انجام شده در این گزارش، می‌توان امکان استفاده از الگوریتم‌های پیچیده‌تر در پردازش سیگنال‌ها را نام برد. در این پروژه از شبکه‌های عصبی پیچیده<sup>۱۵</sup> بهره گرفته شده است. عیب این سیستم نیز افزایش هزینه‌های سخت‌افزاری برای خرید قطعات مربوط به بخش پردازش مرکزی است.

<sup>13</sup> Edge-Node Processing

<sup>14</sup> Cloud Processing

<sup>15</sup> CNN

## فصل دوم

### روش انجام پروژه



## ۲-۱- مقدمه

سامانه‌ی طراحی شده برای تشخیص عیوب پمپ‌های سانرفیوژی با استفاده از یک ماژول شتاب‌سنج، ارتعاشات پمپ را سنجیده و از آن نمونه برداری می‌کند. سپس این سیگنال‌ها با ارتباط بی‌سیم به بخش پردازش مرکزی ارسال شده و وارد یک شبکه‌ی عصبی می‌شود و این شبکه پیش‌بینی می‌کند که سیستم در چه وضعیت سلامتی قرار دارد. در نهایت این اطلاعات در یک داشبورد به صورت برخط برای کاربر به نمایش در می‌آید.

در گام اول یک شبکه‌ی عصبی پیچیده یک بعدی طراحی و با داده‌های موجود در اینترنت تست شد تا عملکرد کلی آن مورد ارزیابی واقع شود. در گام بعدی، دستگاه داده‌برداری شامل یک ماژول شتاب‌سنج، میکروکنترلر، باتری و فرستنده‌ی بی‌سیم طراحی شد تا فرم نهایی این دستگاه تکمیل شود. سپس عملیات داده برداری از پمپ موجود در آزمایشگاه به وسیله‌ی این دستگاه انجام گرفت و ساختار شبکه‌ی عصبی نهایی با استفاده از این داده‌ها اصلاح و سپس آموزش داده شد و عملکرد آن مورد آزمایش قرار گرفت. همچنین با بهره‌گیری از قابلیت‌های آنلاین این سامانه، سناریویی برای تشخیص عیوب جدید به وسیله‌ی این سامانه طراحی شده است. در نهایت بخش پردازش مرکزی راه اندازی شد و با پیاده سازی مدل شبکه‌ی عصبی بر روی آن، زنجیره‌ی کاری سیستم تکمیل گردید.

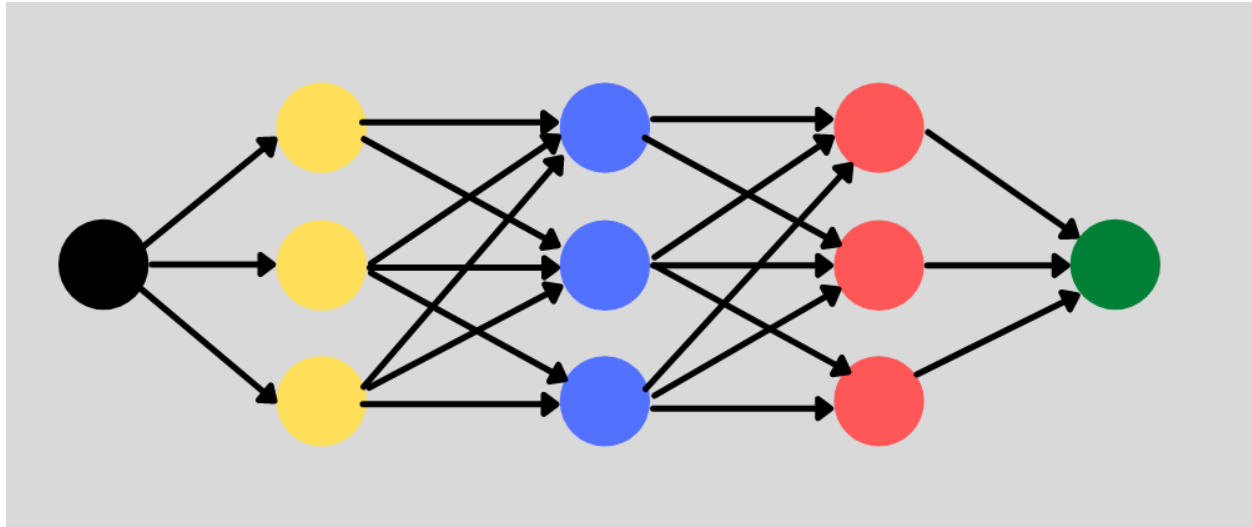
در این بخش ابتدا توضیحاتی در مورد شبکه‌های عصبی داده می‌شود، سپس به فرایند جمع‌آوری داده‌ها از ستاپ آزمایشگاهی و آموزش شبکه پرداخته می‌شود و نتایج آن مورد بحث قرار می‌گیرد. پس از آن سناریویی برای تشخیص عیوب جدید معرفی می‌شود و در نهایت نیز بخش پیاده سازی سخت افزاری این زنجیره انجام می‌گیرد.

## ۲-۲- شبکه‌های عصبی

انواع مختلفی از شبکه‌های عصبی وجود دارد که می‌توان از آن‌ها در پروژه‌های یادگیری ماشین استفاده کرد. شبکه‌های عصبی بازگشتی، شبکه‌های عصبی پیشرو، شبکه‌های عصبی مدولار و موارد دیگر وجود دارد. شبکه‌های عصبی پیچیده (کانولوشن) نوع دیگری از شبکه‌های عصبی متداول است.

### ۲-۲-۱- شبکه‌ی عصبی

شبکه‌های عصبی از مغز ما الگو گرفته‌اند. گره‌های منفردی وجود دارند که لایه‌ها را در شبکه تشکیل می‌دهند و دقیقاً مانند نورون‌های مغز ما مناطق مختلف را به هم متصل می‌کنند.

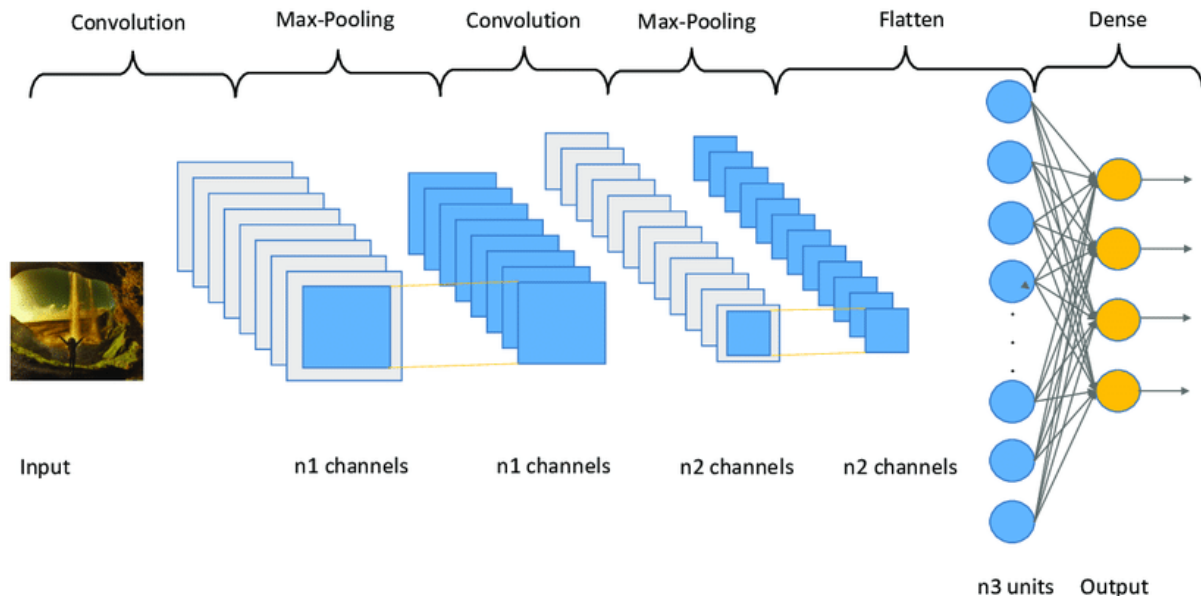


شکل ۲-۱: تصویر شماتیک یک شبکه‌ی عصبی ساده

گره‌های یک لایه، وزنی به آن‌ها اختصاص می‌دهد که تأثیری را که پارامتر بر نتیجه پیش‌بینی کلی دارد تغییر می‌دهد. شبکه‌ی عصبی تمام داده‌های آموزش را در لایه ورودی می‌گیرد؛ سپس داده‌ها را از میان لایه‌های پنهان عبور داده و مقادیر را بر اساس وزن هر گره تغییر شکل می‌دهد و در نهایت مقدار آن را در لایه خروجی برمی‌گرداند.

### ۲-۲-۲- شبکه‌ی عصبی پیچیده

شبکه عصبی کانولوشن نوع خاصی از شبکه عصبی با چندین لایه است. داده‌هایی را که دارای آرایش شبکه‌ای هستند پردازش می‌کند و سپس ویژگی‌های مهم را استخراج می‌کند. یک مزیت بزرگ استفاده از CNN این است که نیازی به انجام بسیاری از مراحل پردازش روی تصاویر و سیگنال‌ها نیست.



شکل ۲-۲: تصویر شماتیک یک شبکه‌ی عصبی پیچیده مورد استفاده در تشخیص تصاویر

فیلترها معمولاً توسط یک مهندس بر اساس ابتکارات ایجاد می‌شوند. CNN می‌تواند بیاموزد مهم‌ترین ویژگی در فیلترها چیست. این امر باعث صرفه‌جویی زیادی در وقت و آزمایش و خطا می‌شود زیرا ما به پارامترهای زیادی احتیاج نداریم.

هدف اصلی الگوریتم شبکه عصبی کانولوشن، تبدیل داده‌ها به فرمی است پردازش آن‌ها ساده تر باشد، بدون آنکه ویژگی‌های اصلی خود را از دست بدهند. این امر همچنین آن‌ها را گزینه‌هایی عالی برای مدیریت مجموعه داده‌های عظیم تبدیل می‌کند.

یک تفاوت بزرگ بین CNN و یک شبکه عصبی معمولی این است که CNN ها از کانولوشن برای مدیریت ریاضیات پشت‌صحنه استفاده می‌کنند. حداقل در یک لایه‌ی CNN به جای ضرب ماتریسی از یک کانولوشن استفاده می‌شود. کانولوشن‌ها دو تابع را به عنوان ورودی دریافت کرده و یک تابع را برمی‌گردانند.

در اصل CNN ها با اعمال فیلتر روی داده‌های ورودی کار می‌کنند. آنچه آن‌ها را بسیار خاص می‌کند این است که این شبکه‌ها می‌توانند فیلترها را در حین اینکه آموزش می‌بینند، تنظیم کنند. به این ترتیب حتی وقتی مجموعه داده‌های عظیمی مانند تصاویر وجود داشته باشد، نتایج به صورت آنی بهبود پیدا می‌کنند.

از آنجاکه می‌توان فیلترها را برای آموزش بهتر CNN به‌روز کرد، این امر نیاز به فیلترهای دستی را از بین می‌برد و به ما در تعداد فیلترهایی که می‌توانیم برای مجموعه‌ی داده‌ها اعمال کنیم و مرتبط بودن آن فیلترها، انعطاف‌پذیری بیشتری می‌دهد. با استفاده از این الگوریتم می‌توانیم روی مشکلات پیچیده‌تری مانند تشخیص چهره کار کنیم.

یکی از مواردی که باعث می‌شود در بعضی از مسائل از CNN ها استفاده نشود، کمبود داده است. هر چند که این شبکه‌ها را می‌توان با حدود ده هزار نمونه نیز آموزش داد، اما هرچه اطلاعات بیشتری در دسترس باشد، CNN بهتر تنظیم می‌شود. این داده‌ها باید تمیز و دارای برچسب باشند تا CNN بتواند از آن‌ها استفاده کند.

شبکه‌های عصبی کانولوشنال بر پایه‌ی یافته‌های علوم اعصاب هستند. آن‌ها از لایه‌های نورون مصنوعی به نام گره ساخته شده‌اند. این گره‌ها توابعی هستند که مجموع وزنی ورودی‌ها را محاسبه می‌کنند و یک نقشه فعال ساز<sup>۱۶</sup> را برمی‌گردانند. این قسمت، بخش کانولوشن شبکه‌ی عصبی است.

هر گره در یک لایه با مقادیر وزنی خود تعریف می‌شود. وقتی تعدادی داده به یک لایه وارد شود، مانند یک تصویر، آن لایه مقادیر پیکسل‌ها را می‌گیرد و برخی از ویژگی‌های بصری را انتخاب می‌کند.

هنگام کار با داده‌ها در یک شبکه‌ی CNN، هر لایه یک نقشه فعال‌سازی را برمی‌گرداند. این نقشه‌ها به ویژگی‌های مهم مجموعه‌ی داده اشاره دارند. اگر به CNN تصویری داده باشید، ویژگی‌های مبتنی بر مقادیر پیکسل (مانند رنگ‌ها) را پیدا می‌کند و یک تابع فعال‌سازی را بر می‌گرداند.

## ۲-۳- تهیه‌ی مجموعه‌ی داده‌ها

شبکه‌های عصبی پیچیده (CNN ها) به دسته‌ای از انواع روش‌های یادگیری ماشین تعلق دارند که به آن‌ها "روش‌های یادگیری تحت نظارت"<sup>۱۷</sup> گفته می‌شود. این اسم از این رو به این دسته از روش‌ها اطلاق می‌شود که برای آموزش آن‌ها علاوه بر داده‌های ورودی، باید برچسب هر داده نیز مشخص شده باشد و به همراه آن داده، به عنوان ورودی به شبکه داده شود.

همان‌طور که پیش‌تر اشاره شد، با توجه به مزیت‌های نسبی استفاده از سیگنال‌های ارتعاشی برای پایش وضعیت سیستم دوار، تصمیم گرفته شد تا از این نوع داده در این پروژه استفاده شود. هم‌چنین به عنوان بستر آزمایش‌ها، از ستاپ آزمایشگاهی شامل یک پمپ آب کوپل شده با موتور تک‌فاز یک اسب بخاری، مخزن آب، روتامتر، فشارسنج و دو عدد شیر که مسئولیت کنترل دبی جریان را بر عهده دارند، استفاده شده‌است. این مجموعه در شکل ۲-۳ نمایش داده شده‌است.

<sup>16</sup> Activation map

<sup>17</sup> Supervised Learning methods



شکل ۲-۳: بستر آزمایشگاهی حلقه-بسته مورد استفاده در پروژه

برای اندازه‌گیری ارتعاشات پمپ، از یک ماژول شتاب‌سنج از نوع LIS3DSH استفاده شده‌است که بر روی قسمت خروجی پمپ نصب می‌شود. این ماژول شتاب‌سنج علاوه بر داده‌برداری با فرکانس ۱۰۶ کیلوهرتز، دارای فیلتر داخلی با فرکانس قطع ۸۰۰ هرتزی است که از پدیده‌ی aliasing جلوگیری می‌کند. علی‌رغم امکان سنجش شتاب در سه محور، به دلیل عواملی چون کمبود حافظه‌ی میکروکنترلر، بالا رفتن تعداد پارامترهای مدل شبکه‌ی عصبی و به تبع افزایش زمان آموزش آن، تصمیم به استفاده از داده‌های شتاب در یک محور شده‌است. هم‌چنین با توجه به دقت بسیار بالای نتایج، نیازی به استفاده از داده‌های شتاب در دو محور دیگر نیست.

در مرحله‌ی جمع‌آوری داده‌ها برای آموزش شبکه‌ی عصبی، داده‌برداری در زمان‌های طولانی در دبی‌های مختلف و شرایط سلامت مختلف جمع‌آوری شد و داده‌های حاصل از میکروکنترلر به کامپیوتر منتقل شد.

همان‌طور که پیش‌تر اشاره شد، برای آموزش شبکه‌های عصبی، به تعدادی نمونه و برجسب‌های متناظر با هر نمونه نیاز داریم. در این مدل، هر نمونه از ۱۰۲۴ داده تشکیل شده‌است و منظور از برجسب نیز یک عدد از ۰ تا ۵ است (در مجموع ۶ عدد، که برابر با تعداد حالت‌های در نظر گرفته‌شده برای سیستم است) که مشخص می‌کند آن نمونه به کدام دسته تعلق دارد.

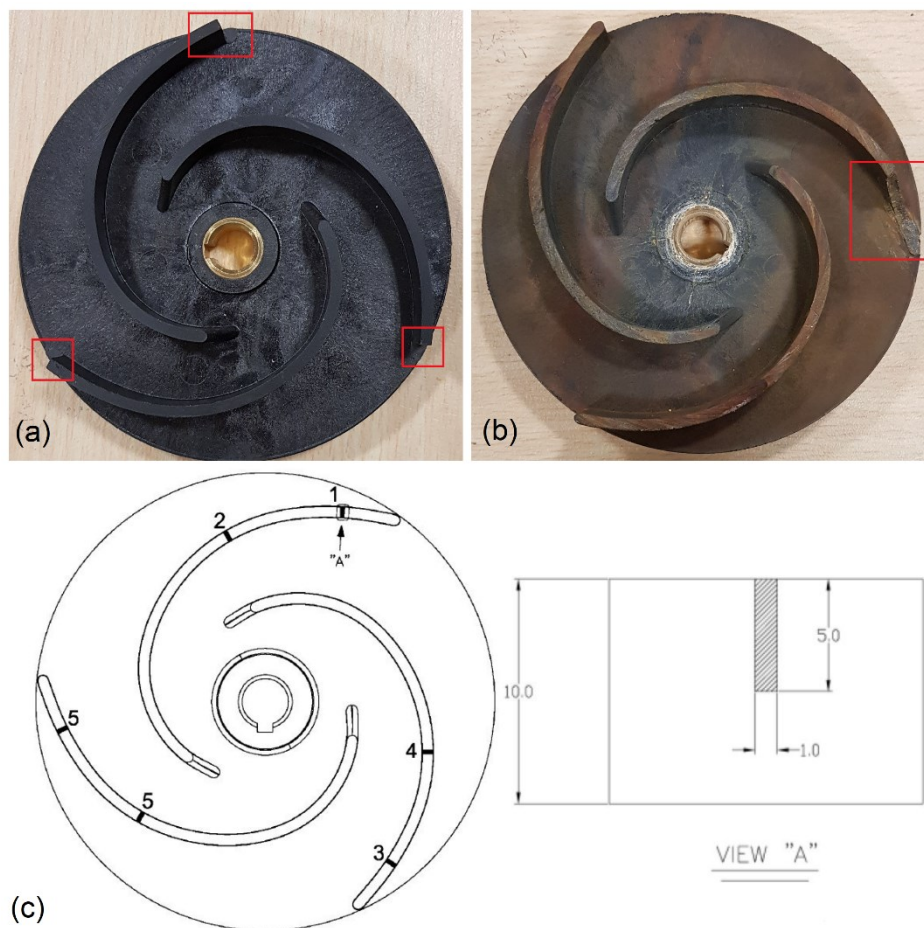
در جدول ۱۲-۱، مجموعه‌ی داده‌های ما برای این آموزش به صورت خلاصه گردآوری شده‌است:

جدول 1۲ :- مجموعه‌ی داده‌های مورد استفاده در فرایند آموزش و تست شبکه‌ی عصبی

نوع داده	فاز آموزش	فاز تست	دبی (لیتر بر دقیقه)	برچسب
سالم	۱۴۶۰	۳۶۰	۱۷۰ → ۲۲۰	۰
دبی پایین	۱۱۲۰	۲۸۰	۸۰ → ۱۲۰	۱
کاویتاسیون	۱۴۴۰	۳۶۰	۱۲۰ → ۱۷۰	۲
Major defect	۱۴۴۰	۳۶۰	۱۵۰ → ۲۰۰	۳
Minor defect	۱۰۴۰	۲۶۰	۱۳۰ → ۱۶۰	۴
ترک	۱۲۸۰	۳۲۰	۱۵۰ → ۲۰۰	۵

همان‌طور که مشاهده می‌شود، در مجموع ۶ حالت سلامتی برای سیستم داریم. در حالت سالم، دبی پمپ عددی بین ۱۷۰ تا ۲۲۰ لیتر بر دقیقه در نظر گرفته شده‌است. حالت دبی پایین، موقعیتی است که پمپ با دبی‌هایی میان ۸۰ تا ۱۲۰ لیتر بر دقیقه کار می‌کند. سیستم در صورت کار کردن مداوم در این حالت، دچار مشکل می‌شود؛ بنابراین، این حالت نیز یک عیب تلقی شده‌است. عیب بعدی، کاویتاسیون است که یکی از جدی‌ترین عیوب پمپ‌هاست و می‌تواند به مرور زمان آسیب جدی به پره‌های آن وارد کند. داده‌های حالت کاویتاسیون نیز در دبی‌های ۱۲۰ تا ۱۷۰ لیتر بر دقیقه گردآوری شده‌است. این عیب به‌وسیله‌ی کنترل جریان دو شیر نصب شده بر سر راه جریان، شبیه‌سازی شده‌است.

سه عیب بعدی، مربوط به حالت‌های خرابی پره است. برچسب‌های ۳ و ۴، به ترتیب مربوط به بریدگی نوک پره در دو اندازه بزرگ و کوچک است. حالت آخر نیز شبیه‌سازی ترک خوردگی پره است که با ایجاد شیارهایی با ابعاد ریز، محقق شده‌است. تصویر این عیوب مربوط به پره در شکل ۲-۴ نشان داده شده‌است.



شکل ۲-۴: (a) عیب پریدگی نوک پره در اندازه‌ی کوچک – (b) عیب پریدگی نوک پره در اندازه‌ی بزرگ – (c) شبیه‌سازی ترک در پره

در جدول مجموعه‌ی داده‌ها، دیده می‌شود که داده‌ها به دو قسمت تقسیم شده‌اند. در فرایند آموزش یک شبکه‌ی عصبی، باید تعدادی از نمونه‌ها (در حدود ۸۰ درصد) به همراه برچسب‌های خود به عنوان ورودی به شبکه داده شوند و تعداد کمتری از داده‌ها (در حدود ۲۰ درصد) بدون برچسب برای سنجش عملکرد شبکه بر روی داده‌هایی که تا به حال ندیده‌است، به کار روند.

برای رسیدن به شبکه و پیش‌پردازش مطلوب، تست‌های متعددی بر روی داده‌ها انجام شد. معمولاً پیش از آموزش یک شبکه‌ی عصبی، مراحل مختلفی از پیش‌پردازش بر روی داده‌های ورودی انجام می‌گیرد. عملیات‌های ریاضی مختلفی مانند هم مقیاس کردن داده‌ها، نرمال‌سازی، تبدیل فوری‌ی سریع و ...، نمونه‌هایی از این عملیات‌ها هستند. از آن‌جا که داده‌های ما همه از نوع شتاب هستند و با توجه به سنسور انتخابی در محدوده‌ی  $\pm 2g$  قرار

دارند، عملاً هم مقیاس هستند. هم‌چنین با اعمال کردن نرمالیزاسیون و تبدیل فوریه‌ی سریع<sup>۱۸</sup> بر روی داده‌ها، تغییرات محسوسی در نتایج تست شبکه‌ی عصبی صورت نگرفت. از آن‌جا که هر عمل ریاضی بر روی داده‌ها نیازمند توان پردازشی است و در نظر گرفتن این نکته که این هزینه‌ی اضافی در مدل ما تفاوت خاصی ایجاد نمی‌کند، مرحله‌ی پیش‌پردازش از فرایند حذف می‌شود.

## ۲-۴- مدل به کار گرفته شده در این پروژه

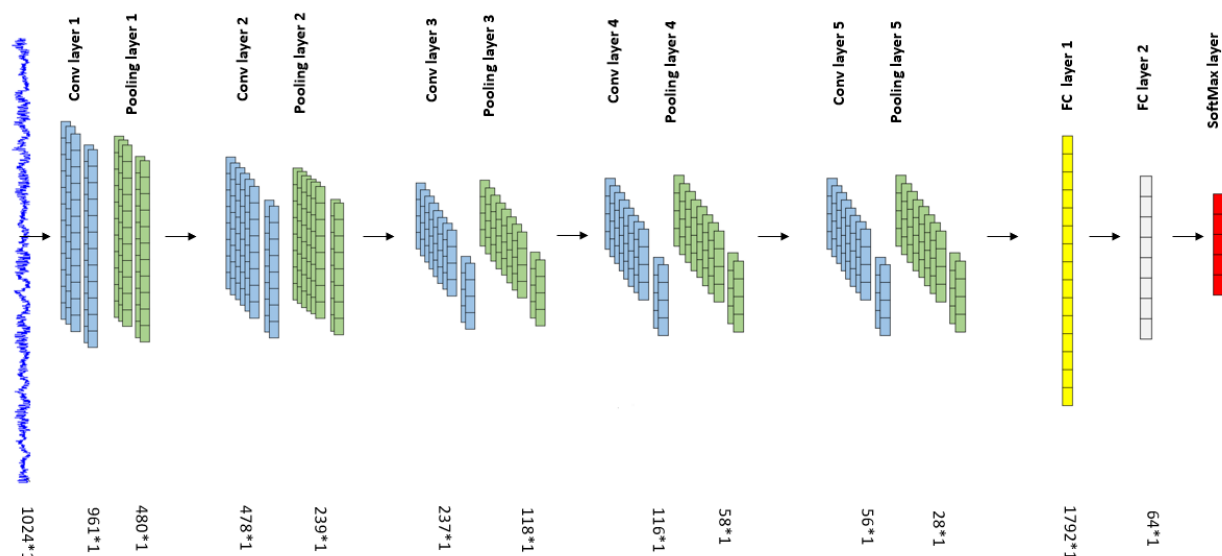
در پروژه‌ی تشخیص عیب پمپ‌های سانتریفیوژی، از یک مدل CNN یک بعدی با کرنل عریض در لایه‌ی اول استفاده شده‌است. در معماری این شبکه، از ۵ لایه‌ی کانولوشن به همراه ۵ لایه‌ی max-pooling بهره گرفته‌ایم تا بتوانیم علاوه بر تشخیص روابط پیچیده موجود در یک سیگنال ارتعاشی، تعداد پارامترهای موجود در این شبکه و در نتیجه زمان لازم برای آموزش آن را تا حد ممکن پایین بیاوریم. کرنل عریض در لایه‌ی اول، شبکه را در برابر نویزهای محیط صنعتی مقاوم می‌کند و نتایج پایدارتری را در شرایط مختلف به ما می‌دهد.

ورودی این شبکه، یک مجموعه شامل ۱۰۲۴ داده‌ی خروجی شتاب‌سنج است. این ورودی‌ها وارد شبکه می‌شود و از فیلترهای مختلفی عبور می‌کند و نهایتاً در خروجی (لایه‌ی soft-max)، به صورت شش عدد ظاهر می‌شود که احتمال وقوع هر یک شش حالت سلامتی سیستم را نمایش می‌دهد.

در شکل ۲-۵، معماری کلی شبکه‌ی طراحی شده نشان داده شده و اندازه‌ی خروجی هر لایه نیز در پایین هر فیلتر نوشته شده‌است:

<sup>18</sup> FFT





شکل ۲-۵: شماتیک معماری شبکه‌ی طراحی شده‌ی نهایی مورد استفاده در پروژه

## ۲-۵- فرایند آموزش شبکه‌ی عصبی

پس از آماده‌سازی داده‌ها و تقسیم آن‌ها به دو دسته برای آموزش و تست، به فرایند آموزش شبکه‌ی عصبی می‌پردازیم. در این فرایند، از زبان برنامه‌نویسی پایتون استفاده شده‌است. کتابخانه‌های متن‌باز موجود در زمینه‌ی شبکه‌های عصبی، سرعت مناسب و بسیاری از ویژگی‌های دیگر، این زبان را به انتخاب اول محققین هوش مصنوعی در سراسر دنیا تبدیل کرده‌است.

همچنین از کتابخانه‌های کارآمد و قدرتمندی چون تنسورفلو<sup>۱۹</sup> و سای کیت لرن<sup>۲۰</sup> نیز استفاده شده‌است. در این کتابخانه‌ها، بسیاری از توابع مورد نیاز تعریف شده‌اند و با کمک آن‌ها، تعریف لایه‌های شبکه‌ی عصبی و عملیات‌های پیچیده‌ی ریاضی بسیار ساده‌تر و بهینه‌تر انجام می‌شود.

### ۲-۵-۱- کتابخانه‌های مورد استفاده

#### کتابخانه‌ی TensorFlow

تنسورفلو یک کتابخانه رایگان و متن‌باز برای برنامه‌نویسی جریان داده و برنامه‌نویسی متمایزگر جهت انجام طیف وسیعی از وظایف است. این کتابخانه برای ریاضیات نمادین محسوب می‌شود و کاربردهای گوناگونی در یادگیری ماشین دارد که از آن جمله می‌توان به پیاده‌سازی شبکه‌های عصبی اشاره کرد. این کتابخانه توسط تیم Google Brain، برای مصارف داخلی گوگل توسعه داده شده بود؛ ولی در نهم نوامبر سال ۲۰۱۵ با گواهینامه آپاچی ۲.۰

<sup>۱۹</sup> TensorFlow

<sup>۲۰</sup> Scikit Learn

متن باز منتشر شد. در حال حاضر، کتابخانه تنسورفلو، در گوگل هم برای پروژه‌های تحقیقاتی و هم پروژه‌های عملیاتی مورد استفاده قرار می‌گیرد. [14]

### کتابخانه‌ی Scikit Learn

سای کیت لرن یک کتابخانه رایگان با زبان پایتون برای یادگیری ماشین است. این کتابخانه دارای الگوریتم‌های یادگیری ماشین مانند ماشین بردار پشتیبان، درخت تصمیم‌گیری و  $k$  نزدیک‌ترین همسایه است؛ همچنین از کتابخانه‌های عددی و آماری پایتون مانند NumPy و SciPy پشتیبانی می‌کند و می‌تواند یادگیری نظارت‌شده و یادگیری بدون نظارت را با کیفیت خوبی انجام دهد. [15]

### کتابخانه‌ی Keract

استخراج وزن لایه‌های شبکه عصبی: برای درک بهتر و دقیق‌تر روند آموزش شبکه عصبی و بررسی اتفاقاتی که در هر لایه از شبکه عصبی رخ می‌دهد، لازم است تا وزن لایه‌های مختلف شبکه عصبی استخراج شود. کتابخانه Keract که یکی از کتابخانه‌های منتشرشده در زبان برنامه‌نویسی پایتون می‌باشد، امکان استخراج وزن لایه‌های یک شبکه عصبی را فراهم می‌کند. [16]

### ۲-۵-۲- آموزش شبکه‌ی عصبی

پس از طی مراحل قبلی و نصب کتابخانه‌های لازم، فرایند آموزش شبکه آغاز می‌شود. داده‌ها به دو قسمت تقسیم شده و معماری شبکه (تعداد لایه‌ها و فیلترها) بر اساس طرحی که پیش‌تر اشاره شد، تعریف می‌شود و با تعریف پارامترهای لازم، فاز آموزش شبکه به پایان می‌رسد.

برای سنجش موفقیت و کارایی یک مدل آموزش دیده‌شده، معمولاً دو روش به کار گرفته می‌شود: ۱- ارزیابی براساس فرضیاتی که باید مدل در آن‌ها صدق کند. ۲- ارزیابی براساس کارایی مدل در پیش‌بینی وضعیت سیستم، با استفاده از داده‌های تست (که در فرایند آموزش از آن‌ها استفاده نشده‌است).

برای بررسی و ارزیابی مدل براساس رویه‌ی شماره یک، تکیه بر داده‌هایی است که مشاهده شده و در آموزش مدل به کار رفته‌اند. برای مثال در رگرسیون خطی، فرض بر این است که باقی‌مانده‌های مدل رگرسیونی باید تصادفی و با واریانس ثابت باشند. همچنین توزیع آن‌ها نیز باید نرمال باشد. بررسی صحت این فرضیات می‌تواند به عنوان معیاری برای سنجش اعتبار مدل محسوب شود. پارامترهای مدل رگرسیونی با کمینه سازی مربعات خطا حاصل می‌شود؛ بنابراین انتظار می‌رود که مدل ساخته شده نسبت به هر مدل دیگری مجموع کمترین مجموع مربعات خطا را نیز داشته باشد. مشخص است که بررسی موارد بالا براساس داده‌هایی که مدل براساس آن ساخته شده

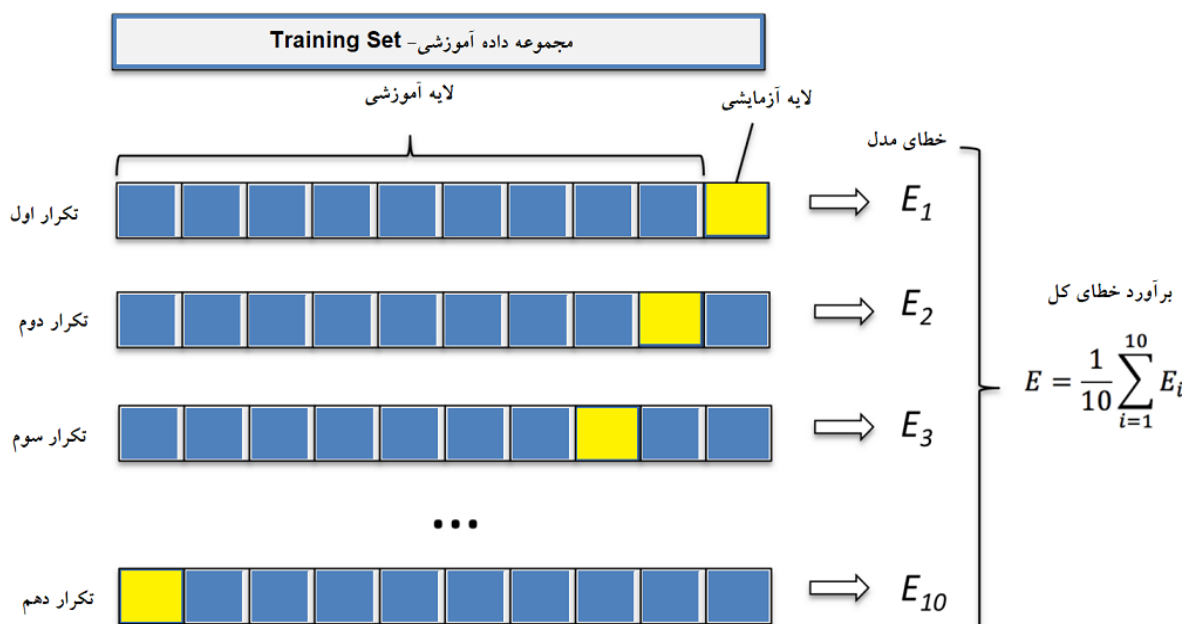
است، میسر است ولی نمی‌توان کارایی مدل را برای داده‌های جدیدی که هنگام مدل‌سازی مشاهده نشده‌اند، سنجید.

اما در معیار دوم سنجش کارایی، داده‌های جدید را به عنوان ورودی به شبکه داده و خروجی آن‌ها را با برچسب‌هایی که مقادیر آن‌ها را از قبل می‌دانیم، مقایسه می‌کنیم. به این ترتیب برای اندازه‌گیری کارایی مدل و بهینه بودن آن متوسط به برآورد خطای مدل براساس داده‌هایی می‌شویم که برای اعتبارسنجی متقابل کنار گذاشته شده‌اند. برآورد این خطا را معمولاً «خطای خارج از نمونه» (Out-of-sample error) می‌نامند.

روش‌های بسیار زیادی برای این منظور وجود دارد اما یکی از بهترین و پرکاربردترین آن‌ها روش K-fold است که در ادامه بررسی می‌شود.

### ۲-۵-۳ روش K-fold

اگر مجموعه‌ی داده‌ها را به طور تصادفی به  $k$  بخش یکسان افراز کنیم، می‌توان در هر مرحله از فرایند آموزش و تست، تعداد  $k-1$  از این دسته‌های داده را به عنوان مجموعه‌ی داده‌ی آموزش و یکی را به عنوان مجموعه‌ی داده‌ی تست در نظر گرفت. شکل زیر، مراحل روش K-Fold را به خوبی نشان می‌دهد. مشخص است که با انتخاب  $k = 10$ ، تعداد تکرارهای این فرایند برابر با ۱۰ خواهد بود و دستیابی به مدل مناسب به سرعت امکان‌پذیر می‌شود. [17]



شکل ۲-۶: فرایند اعتبار سنجی مدل شبکه‌ی عصبی طراحی شده توسط الگوریتم K-fold

در این پروژه روش K-Fold به کمک کتابخانه scikit learn پیاده سازی و شبکه‌ی عصبی اعتبارسنجی شد.

## ۲-۶- به روز رسانی شبکه‌ی عصبی و تشخیص عیوب جدید

یکی از ایرادات روش‌های یادگیری تحت نظارت آن است که پس از طی شدن مرحله‌ی آموزش، تنها قادر هستند تا همان لیب‌هایی را که بر اساس آن‌ها آموزش دیده‌اند، تشخیص دهند و قادر به یادگیری بر اساس مشاهدات جدید نیستند. این امر اگر چه در مواردی مانند تشخیص دست‌خط یا شماره‌ی پلاک و ... مشکل خاصی ایجاد نمی‌کند (چرا که مجموعه‌ی حالات ممکن، معین و محدود است) اما در زمینه‌هایی چون تعمیر و نگهداری پیش‌بینانه، باعث ایجاد مشکلات جدی است.

در یک تجهیز صنعتی دوار مانند پمپ، انواع بسیار زیادی از مشکلات ممکن است روی دهد. تعدادی از این مشکلات در بخش مربوط به جمع‌آوری داده‌ها برای آموزش شبکه‌ی عصبی بحث شده است. علاوه بر مشکلاتی چون کلویتاسیون، دبی پایین جریان ورودی، ترک خوردگی و پدیدگی نوک پره‌ی پمپ، مشکلات دیگری چون عیب بلبرینگ (که خود شامل انواعی از جمله عیب ساچمه، عیب پوسته‌ی داخلی و یا خارجی است)، انواع عیب در موتور پمپ و دیگر مشکلات ناشناخته ممکن است روی دهد. از این رو، ایجاد تمام این عیوب بر روی یک پمپ و تهیه‌ی داده برای آموزش یک شبکه‌ی کامل که همه‌ی این انواع عیب را پوشش دهد، عملاً امری غیر ممکن است.

در این پروژه سعی شده است تا با بهره گرفتن از قابلیت‌های بر خط بودن شبکه و وجود دیتابیس در سیستم پردازش مرکزی، بر این مشکل غلبه کرده و بتوان قابلیت تشخیص عیوب جدید را نیز به سیستم اضافه کرد.

### ۲-۶-۱- روش به کار گرفته شده

برای اینکه بتوان به هدف مورد نظر دست یافت، ابتدا باید شبکه‌ای را طراحی کرد و آموزش داد به گونه‌ای که بتواند عیوبی را که پیش‌تر به آن معرفی شده است، تشخیص دهد؛ این شبکه به طور مفصل مورد بحث قرار گرفت و از همان ساختار در این بخش نیز استفاده خواهد شد.

گام بعدی، یافتن روشی است که بتوان به کمک آن، با دقت بالایی تعیین کرد که عیب متعلق به دسته‌ی جدیدی است که تا به حال به شبکه‌ی عصبی معرفی نشده است. اگر بتوان به این اطمینان رسید، آن‌گاه می‌توان از برخط بودن شبکه استفاده کرد و با جمع‌آوری داده‌های جدید از وضعیت فعلی سیستم (که با توجه به سالم نبودن وضعیت سیستم، قطعاً نوع جدیدی از عیب خواهد بود)، به اندازه‌ای که تعداد این داده‌ها برابر با تعداد داده‌های مربوط به هر کلاس در زمان آموزش شبکه‌ی پیشین شود، شبکه را مجدداً وارد مرحله‌ی آموزش کرد و بدین ترتیب، عیب جدید را به سایر عیوبی که شبکه می‌تواند پیش‌بینی کند، اضافه کرد.

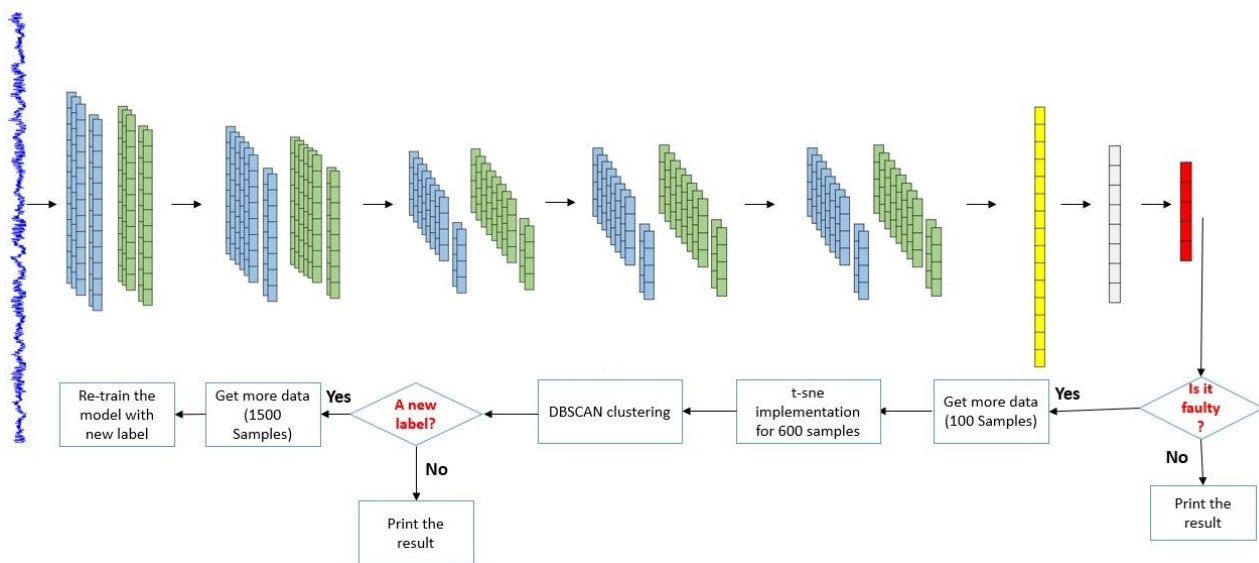
همان‌طور که در شکل ۲-۵ نشان داده شده است، در لایه‌ی ماقبل لایه‌ی خروجی (یعنی لایه‌ی FC2)، ۶۴ نورون وجود دارد. خروجی این نورون‌ها، ویژگی‌های استخراج شده از هر نمونه‌ی ورودی است. به مجموعه‌ی خروجی‌های این ۶۴ نورون می‌توان به دید یک بردار با ۶۴ مولفه نگاه کرد.

پیش‌تر و در بخش مربوط به بصری‌سازی خروجی لایه‌ها، روش t-SNE معرفی شد و عنوان شد که این الگوریتم روش قدرتمندی برای کاهش ابعاد است. در این بخش از این قابلیت روش t-SNE نه برای مصورسازی، که برای کاهش ابعاد بردار ویژگی‌های خروجی اشاره شده در پاراگراف قبلی استفاده می‌شود. در نتیجه‌ی این امر، تعداد مولفه‌های خروجی لایه‌ی FC2 به تنها دو عدد کاهش خواهد یافت که مشخص کننده‌ی یک نقطه در فضای صفحه است.

حال با در نظر گرفتن اینکه خروجی هر داده‌ی وارد شده به شبکه یک نقطه است، ابتدا باید بررسی کرد که آیا هر مجموعه نقاط متعلق به یک وضعیت سلامت سیستم، تشکیل یک دسته‌ی جداگانه خواهد داد یا خیر. در شکل ۳-۲ این موضوع کاملاً تایید شده است. نتیجه‌ای که از این موضوع گرفته می‌شود آن است که اگر عیب جدیدی در سیستم به وقوع بپیوندد، باید در دسته‌ی جدیدی در این صفحه‌ی دو بعدی قرار گیرد.

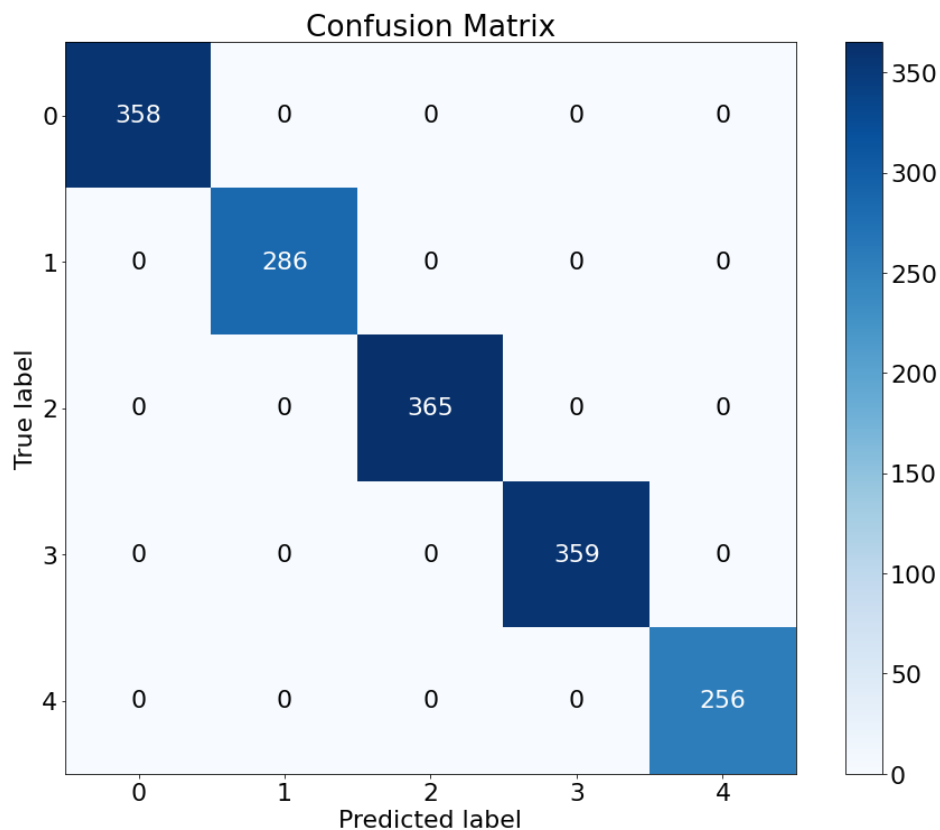
ایجاد شدن دسته‌های متفاوت و متمایز برای هر عیب در فضای دو بعدی، این قابلیت فوق‌العاده را به ارمغان می‌آورد که بتوان با کمک یک الگوریتم خوشه بندی، تعداد این دسته‌ها را تعیین کرد. در این صورت اگر تعداد خوشه‌های شامل داده‌های جدید با تعداد لیبل‌های شناخته‌شده برای شبکه در حالت فعلی آن تفاوت داشت، می‌توان نتیجه گرفت که عیب جدیدی در سیستم ظاهر شده است و باید شبکه را مجدداً آموزش داد.

فرایند کلی این بخش در شکل ۲-۷ آمده است.



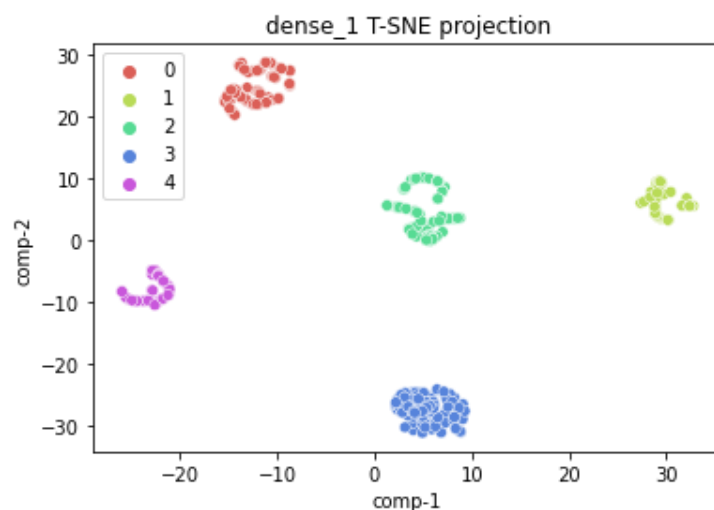
شکل ۲-۷: الگوریتم طی شده برای افزودن عیب جدید به برچسب‌های قبلی

برای آزمودن این روش، آزمایشی شبیه‌سازی شده است. در این آزمایش، از همان مجموعه داده‌های قبلی استفاده می‌شود. با این تفاوت که عیب مربوط به پرهی ترک خورده از ابتدا به سیستم داده نمی‌شود و فرض می‌شود که این عیب پیش از این به سیستم شناسانده نشده است. پس از آموزش وضعیت دیگر سلامت سیستم به شبکه، ماتریس در هم‌ریختگی آن بدین صورت در می‌آید:



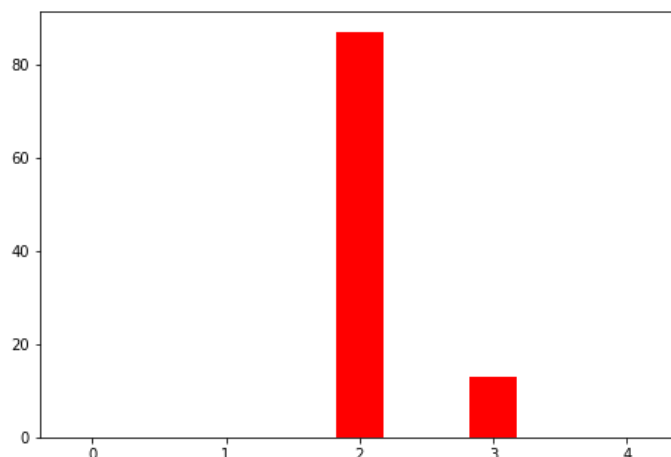
شکل ۲-۸: ماتریس درهم‌ریختگی با پنج کلاس

همچنین بصری سازی خروجی لایه‌ی آخر نیز مطابق شکل ۲-۹ است.



شکل ۲-۹: بصری سازی خروجی لایه‌ی آخر در حالت شبکه‌ی آموزش داده شده با پنج کلاس

در گام بعدی طبق شکل ۲-۷، باید تشخیص داد آیا داده‌ی جدید ورودی به شبکه مربوط به وضعیت سالم سیستم است یا خیر. با وارد کردن داده‌های حالت ترک، شبکه‌ی عصبی پیش‌بینی می‌کند که این داده‌ها بیشتر به کلاس‌های ۲ و ۳ تعلق دارند. این نتیجه در شکل ۲-۱۰ آمده است.



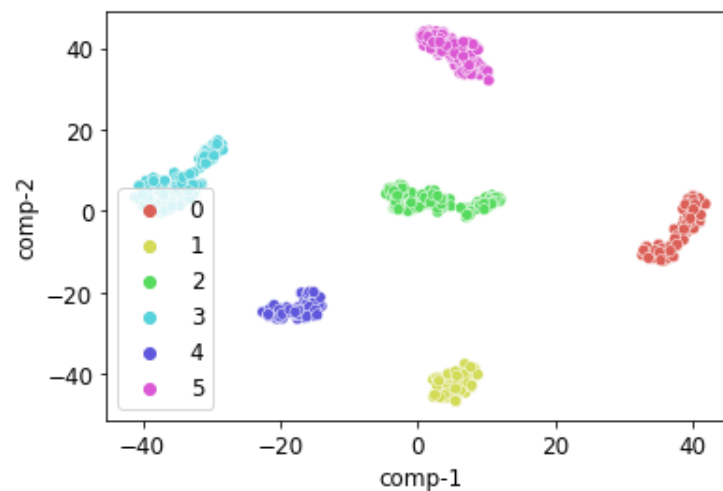
شکل ۲-۱۰: خروجی پیش‌بینی شده برای عیب جدید - تست شده بر روی ۱۰۰ داده

این نتایج نشان می‌دهد که شبکه با مشاهده‌ی نوع جدیدی از یک عیب، خروجی‌های متغیری را پیش‌بینی خواهد کرد. با میانگین گرفتن بر روی چند پیش‌بینی اخیر سیستم (برای مثال ۱۰ داده‌ی ورودی اخیر) مشخص خواهد شد که آیا داده مربوط به حالت معیوب سیستم است یا خیر. در این آزمایش، این امر مشهود است.

حال پس از معلوم شدن معیوب بودن حالت سیستم، باید مشخص شود که آیا این عیب متعلق به دسته‌ی جدیدی است یا خیر. برای خوشه بندی با دقت بالاتر، باید از تعداد داده‌ی بیشتری استفاده شود. در این آزمایش، صد داده از وضعیت فعلی سیستم جمع‌آوری می‌شود تا بتوان آن‌ها را با داده‌های پیشین مقایسه کرد.

برای اینکه شیوه‌ی انجام کار بهتر مشخص شود، با وارد کردن صد داده از هر حالت (۵۰۰ داده مربوط به ۵ کلاس پیشین و ۱۰۰ داده مربوط به وضعیت جدید سیستم) به شبکه‌ی عصبی و ترسیم خروجی کاهش‌یافته‌ی لایه‌ی FC2 توسط روش t-SNE، به تصویر شکل ۲-۱۱ می‌رسیم. فرایند توصیف‌شده در این پاراگراف، برای توضیح بهتر است و خودمان تعیین کرده‌ایم که تعداد کلاس‌ها شش عدد است.





شکل ۲-۱۱: بصری‌سازی خروجی کاهش‌یافته‌ی لایه‌ی FC2 توسط روش t-SNE

در این تصویر مشخص است که داده‌های جدید کاملاً در دسته‌ی جدیدی قرار گرفته‌اند. اما تشخیص این امر باید توسط بخش پردازش مرکزی انجام گیرد و فرایند کاملاً اتوماتیک باشد. بدین منظور، از الگوریتم‌هایی برای دسته‌بندی داده‌های تصویر بالا (که همان خروجی کاهش‌یافته‌ی ۶۰۰ داده است) استفاده کرد.

الگوریتم‌های متفاوتی برای خوشه‌بندی وجود دارند که هر یک مزایا و معایب خاصی دارند. برای افزودن عیب‌های جدید در این پروژه، هر یک از دو الگوریتم DBSCAN و K-Means تست شده و قابل استفاده هستند.

## ۲-۶-۲- الگوریتم DBSCAN:

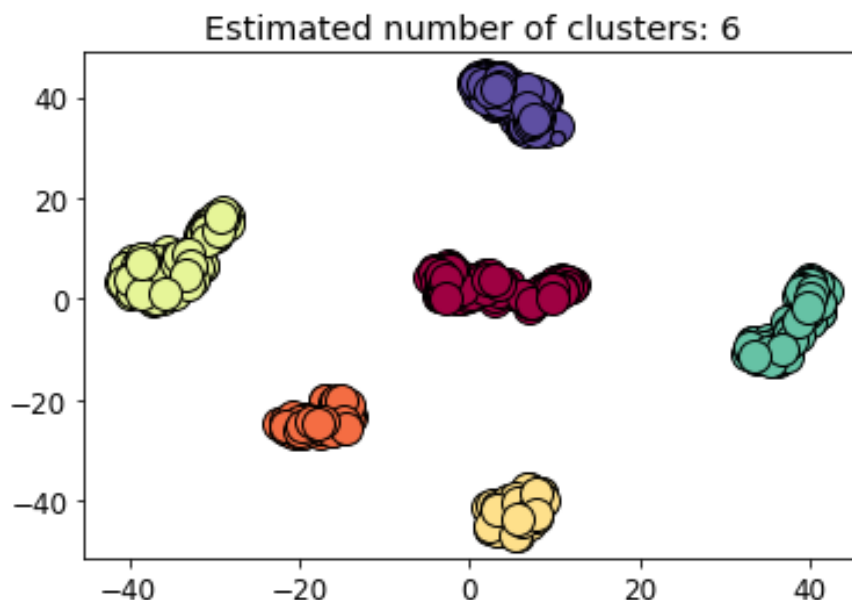
نام کامل این الگوریتم، خوشه‌بندی فضایی مبتنی بر چگالی در کاربردهای دارای نویز<sup>۲۱</sup> است که به اختصار به آن DBSCAN گفته می‌شود. در الگوریتم DBSCAN نیازی به این نیست که تعداد خوشه‌ها از ابتدا تعیین شود. این الگوریتم می‌تواند خوشه‌های دارای اشکال پیچیده را کشف کند. همچنین، می‌تواند نقاط داده‌ای که بخشی از هیچ خوشه‌ای نیستند (نقاط دورافتاده یا ناهنجار) را شناسایی کند. این قابلیت برای تشخیص ناهنجاری بسیار مفید است. DBSCAN با شناسایی نقاطی که در نواحی شلوغ (چگال) از فضای ویژگی<sup>۲۲</sup> قرار دارند کار می‌کند. منظور از نواحی چگال، قسمت‌هایی است که نقاط داده بسیار به یکدیگر نزدیک هستند (نواحی چگال در فضای ویژگی). برخی از نکات کلیدی پیرامون الگوریتم DBSCAN در ادامه آمده‌اند [18].

- دو پارامتر min\_samples و eps در الگوریتم DBSCAN وجود دارد.

<sup>21</sup> Density Based Spatial Clustering of Applications with Noise

<sup>22</sup> Feature Space

- هر نقطه داده، از دیگر نقاط داده فاصله‌ای دارد. هر نقطه‌ای که فاصله‌اش با یک نقطه مفروض کمتر از eps باشد، به عنوان همسایه آن نقطه در نظر گرفت می‌شود.
  - هر نقطه داده مفروضی که min\_samples همسایه داشته باشد، یک نقطه مرکزی<sup>۲۳</sup> محسوب می‌شود.
  - نمونه‌های مرکزی<sup>۲۴</sup> که نسبت به یکدیگر نزدیک‌تر از فاصله eps هستند، در خوشه مشابهی قرار می‌گیرند.
- با تعیین پارامترهای لازم (که بر اساس تجربه و با چند مورد تست کردن شبکه بدست می‌آیند و مقادیر قابل اعتمادی برای ادامه‌ی فرایند هستند)، می‌توان تعداد خوشه‌ها را تعیین کرد. شکل ۲-۱۲ نشان می‌دهد که مجموعه‌ی ۶۰۰ داده به چه صورتی طبقه توسط این روش دسته‌بندی شده‌اند.



شکل ۲-۱۲: خوشه بندی داده‌ها توسط الگوریتم DBSCAN

بنابراین، الگوریتم DBSCAN قادر است تا تعداد خوشه‌ها را به درستی تعیین کند.

## ۲-۶-۳- الگوریتم K-Means

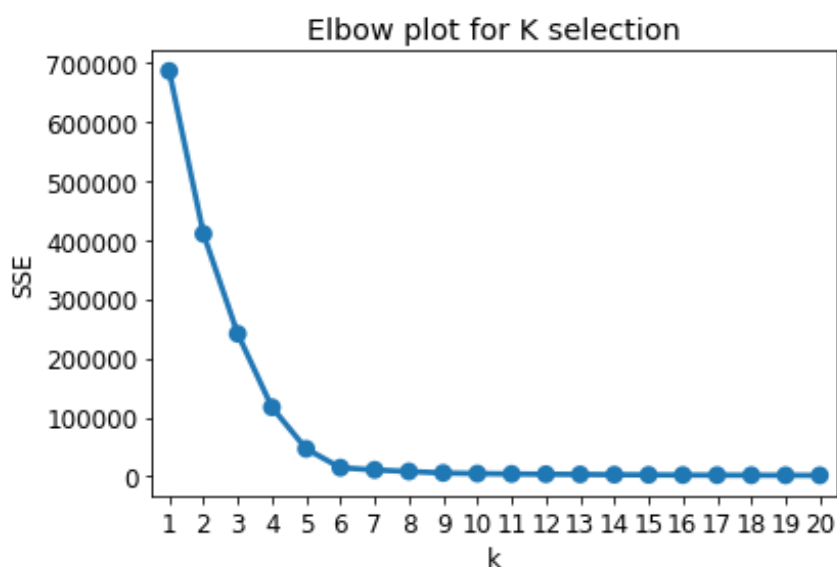
در خوشه‌بندی K-Means یا K-میانیگین، تلاش می‌شود تا مراکز دسته‌ای یافت شوند که نماینده ناحیه خاصی از داده‌ها هستند. در این الگوریتم، هر نقطه داده به نزدیک‌ترین مرکز خوشه نسبت به خودش، تخصیص داده می‌شود.

<sup>23</sup> Core

<sup>24</sup> core samples

سپس، مرکز خوشه‌ها بر اساس میانگین نقاط داده‌ای که به آن خوشه تخصیص داده شده‌اند مجدداً محاسبه و تعیین می‌شوند. این الگوریتم هنگامی پایان می‌یابد که تخصیص نمونه‌ها به خوشه‌ها دیگر تغییر نکند. [19]

برای استفاده از این الگوریتم، توجه به این نکته حائز اهمیت است که باید تعداد خوشه‌ها از ابتدا تعیین شود. مشخص است که تعداد خوشه‌ها باید برابر عددی مساوی یا یک عدد بیشتر از تعداد کلاس‌های فعلی باشد. اما برای مقایسه‌ی نتیجه‌ی خوشه‌بندی با این دو عدد، از معیار جمع مربعات خطا (SSE) استفاده می‌شود که با افزایش تعداد خوشه‌ها، این خطا کم خواهد شد. پس روش بهتر آن است که با تعداد دسته‌های کمتر و بیشتر از این مقادیر نیز خوشه‌بندی را انجام داده و با شناسایی نقطه‌ی زانویی نمودار، تعداد خوشه‌ها تعیین شود.



شکل ۲-۱۳: نمودار جمع مربعات خطا برای دسته‌بندی با تعداد خوشه‌های مختلف توسط الگوریتم K-Means

همان طور که در شکل ۲-۱۳ مشخص است، نقطه‌ی زانویی در این نمودار برابر عدد ۶ است. بنابراین الگوریتم K-Means نیز قادر به تشخیص تعداد خوشه‌های جدید بوده است.

از معایب الگوریتم K-Means می‌توان به عدم توانمندی در دسته‌بندی داده‌هایی که در صفحه به صورت اشکال S مانند یا قطری باشند، اشاره کرد. عیب الگوریتم DBSCAN نیز یافتن پارامترهای اولیه‌ی مورد نیاز است که باید برای هر سیستم، چندین بار آزمایش شود تا مقادیر مناسبی انتخاب شود.

در نهایت و پس از مشخص شدن این موضوع که سیستم با عیب جدیدی مواجه شده است، تعداد بسیار بیشتری داده جمع‌آوری می‌شود تا مجموعه‌ی داده‌های لازم برای آموزش از نظر تعداد انواع حالت‌ها به تعادل برسد. پس از آن نیز شبکه مجدداً آموزش داده شده و قادر به تشخیص عیب جدید نیز خواهد بود.

## ۲-۷- سخت افزار بخش داده برداری

### ۲-۷-۱- شتاب سنج

شتاب سنج حسگری است که شتاب مکانیکی را به سیگنال الکتریکی متناسب با آن تبدیل می کند. شتاب نرخ تغییرات سرعت در واحد زمان می باشد که می تواند ثابت (شتاب استاتیک) یا متغیر (شتاب دینامیک) و یا گذرا باشد. شتاب سنج ابزاری است که شتاب نسبی را اندازه گیری می کند. شتاب نسبی، شتاب (نرخ تغییر سرعت) یک جسم در چارچوب آنی خود است. شتاب نسبی با شتاب مطلق (که شتاب در یک سیستم مختصات ثابت است) تفاوت دارد. [20]

شتاب سنج دارای مدل های یک محوری و چندمحوری است که می توانند اندازه و جهت شتاب را به عنوان یک کمیت برداری اندازه گیری کنند. شتاب سنج برای اندازه گیری ارتعاش، شوک و ضربه به کار می رود؛ ولی می توان از این حسگر برای کاربردهای دیگری نیز استفاده کرد. به کمک انتگرال گیری از شتاب می توان سرعت و جابجایی را اندازه گیری نمود و با کمک شتاب استاتیک می توان زاویه قرارگیری محصول نسبت به محور جاذبه زمین و یا میزان تراز سطح را اندازه گیری نمود.

براساس تکنولوژی بکار رفته در شتاب سنج های متداول، آن ها را به دسته های مختلفی طبقه بندی می کنند که از جمله ی آن ها می توان به شتاب سنج پیزوالکتریک، شتاب سنج های کرنشی و شتاب سنج های جریان گردابی اشاره کرد. شتاب سنج های جدید نیز معمولاً بر پایه ی سیستم های میکروالکترومکانیکی (MEMS) هستند. شتاب سنج های میکرومکانیکی برای اندازه گیری در محدوده های بسیار وسیعی، که به هزاران g هم می رسد، به کار می روند. طراح باید مصالحه ای بین حساسیت و حداکثر مقدار شتاب قابل اندازه گیری انجام دهد.

شتاب سنج ها کاربردهای بسیار گسترده ای دارند و در زمینه های مختلفی از جمله در علوم زمین شناسی و زلزله، زیست شناسی و مطالعه و شناخت الگوهای رفتاری، موبایل ها و هارد دیسک ها، کاربردهای سلامتی (در ساعت ها و مچ بند های هوشمند) و هم چنین در علوم مهندسی و صنعت پایش ساختمان ها و تجهیزات صنعتی استفاده می شود.

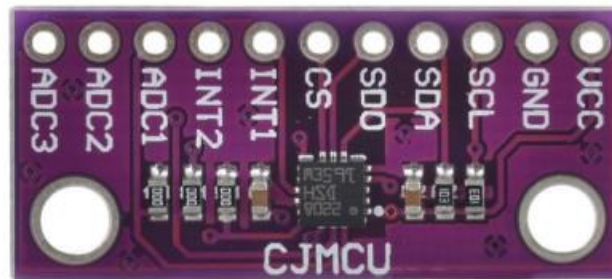
در این پروژه، هدف استفاده از سیگنال های ارتعاشی پمپ سانتریفیوژی برای تشخیص عیب های آن است. سه معیار مهم و اصلی برای انتخاب یک شتاب سنج عبارتند از بازه ی شتاب قابل اندازه گیری، پهنای باند سنسور و پروتکل های ارتباطی مازول شتاب سنج با میکروکنترلر.

اکثر ماشین های دوار موجود با سرعتی بین ۲۰۰ تا ۶۰۰۰ دور بر دقیقه کار می کنند. فرکانس در حداکثر دور (۶۰۰۰ دور بر دقیقه) طبق رابطه ی زیر برابر با ۱۰۰ هرتز است.

$$6000RPM = \frac{6000}{60} \times 2 \times \pi \frac{rad}{s} = \frac{6000}{60} \times 2 \times \pi \times \frac{1}{2 \times \pi} Hz = 100Hz$$

بنابراین برای اندازه‌گیری شتاب ارتعاشات یک ماشین دوار به یک شتاب‌سنج با حداقل پهنای باند ۱۰۰ هرتز احتیاج داریم. اکثر سنسورهای شتاب‌سنج از این پهنای باند به خوبی پشتیبانی می‌کنند. همچنین بازه‌ی شتاب قابل اندازه‌گیری، در حدود  $\pm 2g$  کفایت می‌کند. این قابلیت را نیز بسیاری از شتاب‌سنج‌ها دارند.

بر اساس این ویژگی‌ها، چند ماژول شتاب‌سنج مختلف مورد ارزیابی واقع شدند. فاکتورهایی چون موجود بودن در بازار ایران، قیمت ماژول و همچنین ابعاد آن نیز دارای اهمیت بود. در نهایت، ماژول CJMCU LIS3DSH انتخاب شد که تصویر آن در شکل ۲-۱۴ نمایش داده شده‌است.



شکل ۲-۱۴: ماژول شتاب‌سنج CJMCU LIS3DSH

این ماژول یک شتاب‌سنج دیجیتال است که توانایی اندازه‌گیری شتاب تا بازه‌ی  $\pm 16g$  با پهنای باند حداکثر ۱.۶ کیلوهرتز در سه راستای x، y و z را داراست. همچنین یک فیلتر داخلی با پهنای باند نهایتاً ۸۰۰ هرتزی دارد که باعث جلوگیری از پدیده‌ی aliasing می‌شود. شرکت ST در برخی از بردهای خود از این ماژول استفاده کرده‌است. با بررسی‌های پارامترهای یاد شده و نیز تست‌های بسیار از نظر دقت، توان مصرفی و راحتی در کارکرد، این ماژول برای ادامه کار انتخاب شد. ارتباط با این ماژول از طریق درگاه‌های I2C و SPI فراهم است که در این پروژه از درگاه SPI برای دریافت داده‌های شتاب از این سنسور، استفاده شده‌است.

## ۲-۷-۲- باتری

بخشی از سیستم که بر روی تجهیز دوار نصب می‌شود، باید بدون اتصال به منبع تغذیه‌ی خارجی بتواند به صورت مستقل عمل کند و قابلیت جابجایی داشته‌باشد. بنابراین یک باتری باید توان لازم را برای ماژول شتاب‌سنج، میکروکنترلر و ماژول فرستنده تامین کند. ولتاژ ورودی این قطعات، نهایتاً ۳.۳ ولت است و یک باتری استاندارد با ولتاژ ۳.۷ ولت کاملاً می‌تواند از آن‌ها پشتیبانی کند.

باتری‌های موجود در بازار به دو دسته‌ی قلمی و تخت تقسیم می‌شوند. با توجه به آن که یکی از پارامترهای مهم در طراحی این دستگاه اندازه‌ی آن بود، و افزایش بسیار زیاد عرض باتری‌های تخت در ظرفیت‌های بالا، تصمیم بر آن شد تا از باتری استوانه‌ای استفاده شود.

در نهایت با توجه به قیمت و ظرفیت باتری‌های موجود در بازار، باتری Panasonic NCR18650B با ولتاژ ۳.۷ ولت و ظرفیت ۳۳۵۰ میلی‌آمپر ساعت انتخاب شد.

## ۲-۷-۳- مازول ارتباطی

دستگاه پایش‌گر ارتعاشاتی، در دسته وسایل مبتنی بر شبکه‌ی اینترنت اشیا قرار می‌گیرد و به گونه‌ای طراحی شده‌است که بتواند به صورت بی‌سیم و بر خط، داده‌ها را به مرکز اصلی ارسال کند؛ بدین منظور، این دستگاه نیازمند یک مازول ارتباطی بی‌سیم است. از میان پروتوکل‌های موجود، پروتوکل‌هایی چون وایفای، ترد، زی ویو، بلوتوث و بلوتوث کم مصرف، لورا، سیگفاکس و زیگبی بررسی شدند و در نهایت با توجه به ویژگی‌های ارتباط مورد نیاز مانند مسافت ارتباطی کوتاه، نرخ انتقال داده نه چندان بالا، و همچنین ملاحظات طراحی دستگاه از جمله مصرف باتری پایین و قیمت پایین، پروتوکل ارتباطی زیگبی انتخاب شد. زیگبی یک استاندارد شبکه‌ی بی‌سیم کم هزینه و کم مصرف است که هدف آن توسعه استفاده از دستگاه‌های با طول عمر باتری طولانی در کاربردهای مختلف کنترل و نظارت بی‌سیم می‌باشد. دستگاه‌های زیگبی زمان تاخیر کمی دارند که باعث کمتر شدن جریان متوسط مصرفی می‌شود. چیپ‌های زیگبی عمدتاً به همراه رادیوها و میکروکنترلرهایی که حافظه‌ی فلش بین ۶۰ تا ۲۵۶ کیلوبایت دارند به کار می‌رود. فرکانس ۲.۴ گیگاهرتز متداول‌ترین فرکانس در استانداردهای جهانی است. این فرکانس در چین برابر ۷۸۴ مگاهرتز، در اروپا برابر ۸۵۸ مگاهرتز و در آمریکا و استرالیا برابر ۹۱۵ مگاهرتز می‌باشد. نرخ انتقال داده‌ها از ۲۰ کیلوبیت بر ثانیه در باند ۸۶۸ مگاهرتز تا ۲۵۰ کیلوبیت بر ثانیه در باند ۲.۴ گیگاهرتز متغیر است [21].

برای پیاده‌سازی پروتوکل زیگبی، از مازول ارتباطی XB24-Z7UIT-004 استفاده شده است. تصویر این مازول در شکل ۲-۱۵ ارائه شده است. این مازول ساخت شرکت Digi بوده و با استفاده از پروتکل UART می‌توان با آن ارتباط برقرار کرد. مشخصات کلی این سنسور در جدول ۲۲- ارائه شده است.

جدول ۲۲ :- مشخصات فنی ماژول XBee [22]

Name	Value
RF Family/Standard	802.15.4
Protocol	Zigbee
Modulation	DSSS
Frequency	2.4GHz
Data Rate	250kbps
Power - Output	1dBm
Sensitivity	-96dBm
Interfaces	UART
Antenna Type	U.FL
Supply Voltage	2.1V – 3.6V
Current - Receiving	38mA – 40mA
Current - Transmitting	35mA – 45mA
Mounting Type	Though Hole



شکل ۲- ۱۵: ماژول ارتباطی XBee

همانطور که در جدول ۲۲ - مشخص است، نوع آنتن در این ماژول از نوع U.FL است و در این سیستم از آنتن 2.4GHz 3dBi استفاده شده است که شکل ۲- ۱۶ تصویر آن را نمایش می‌دهد.



شکل ۲-۱۶: آنتن ارتباط بی سیم با فرکانس ۲.۴ گیگاهرتز و بهره‌ی 3dBi

اتصال این آنتن از نوع RPSMA می‌باشد و برای اتصال آن به ماژول مدنظر از تبدیل U.FL به RPSMA استفاده شده است که شکل ۲-۱۷ نمایش داده شده است.



شکل ۲-۱۷: تبدیل U.FL به RPSMA

## ۲-۷-۴- میکرو کنترلر

برای انجام این پروژه، به میکروکنترلری با ویژگی‌های خاص یا توان پردازشی و رم بالایی احتیاج نبود؛ چرا که این میکروکنترلر تنها وظیفه‌ی نمونه برداری را باید انجام دهد و داده‌ها را به ماژول فرستنده‌ی بی‌سیم منتقل کند تا به مرکز پردازشی اصلی برسد. بنابراین، در میان گزینه‌های ارزان موجود در بازار ایران بررسی صورت گرفت و در نهایت، برد توسعه‌ی Raspberry Pi Pico انتخاب شد. این برد توسعه از میکروکنترلر RP2040 به عنوان هسته‌ی مرکزی استفاده می‌کند. تصویر این برد توسعه در شکل ۲-۱۸ نشان داده شده است.





شکل ۲-۱۸: میکروکنترلر رزبری پای پیکو

همچنین در جدول زیر، مشخصات این میکروکنترلر آمده است.

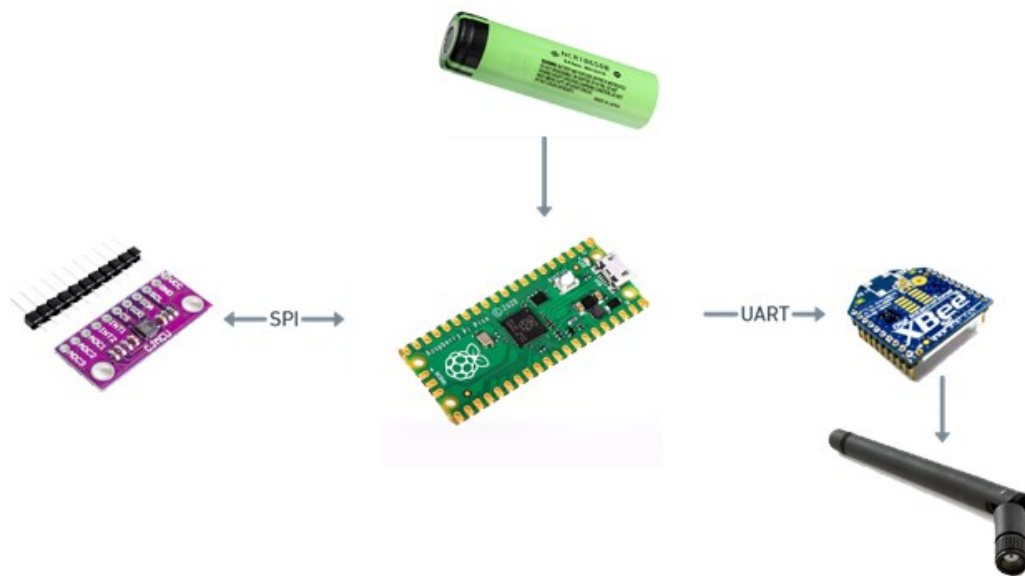
جدول ۳۲ - مشخصات میکروکنترلر RP2040 [23]

Name	Value
Flash Memory	2 Mbyte
CPU Architecture	Dual-core Arm Cortex-M0+
RAM	264 Kbyte
CPU Clock Frequency	133 MHz
ADC	1x12bit, 500KSPS, 5 channels
Timers	1x64bit
Communication Interfaces	2x I <sup>2</sup> C 2x UART 2x SPI USB 1.1

برد رزبری پای پیکو ابعادی بسیار کوچک (تنها ۵۱ در ۲۱ میلی‌متر طول و عرض) و قیمتی بسیار پایین (تنها ۴ دلار) دارد که آن را به گزینه‌ای ایده‌آل تبدیل کرده است. همچنین پشتیبانی از پروتکل‌های متعدد ارتباطی و قابلیت برنامه‌پذیری با زبان‌های C و میکروپایتون، از مزایای دیگر استفاده از این برد توسعه است. میکروپایتون یک پیاده‌سازی ساده و کارآمد از زبان برنامه‌نویسی پایتون ۳ است که شامل زیرمجموعه کوچکی از کتابخانه استاندارد

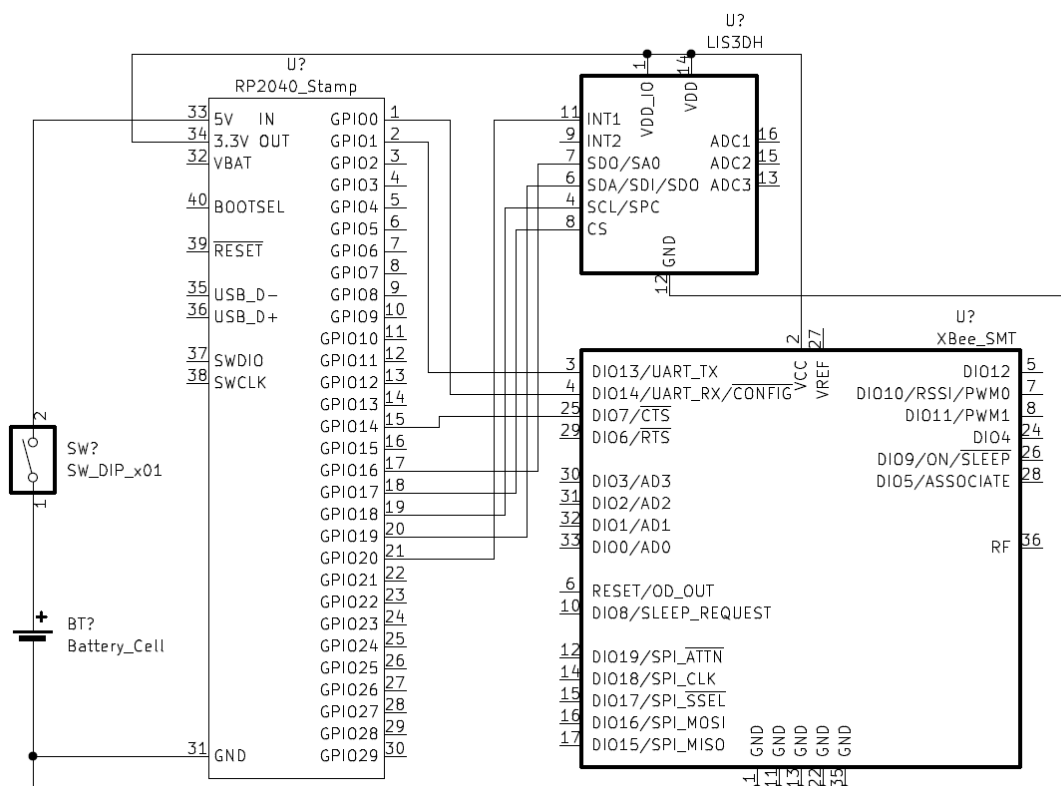
پایتون است و برای اجرا بر روی میکروکنترلرها و در محیط های محدود بهینه شده است و روی ۱۶ کیلوبایت رم نیز قابل اجراست. [24]

در نهایت پس از انتخاب همه ی اجزای مورد نیاز در بخش قابل حمل، شماتیک ارتباطات کلی میان اجزاء به صورت شکل ۲-۱۹ در می آید.



شکل ۲-۱۹: شماتیک ارتباط میان اجزای بخش قابل حمل

شماتیک اتصالات الکترونیکی این قطعات نیز در شکل ۲-۲۰ ارائه شده است.

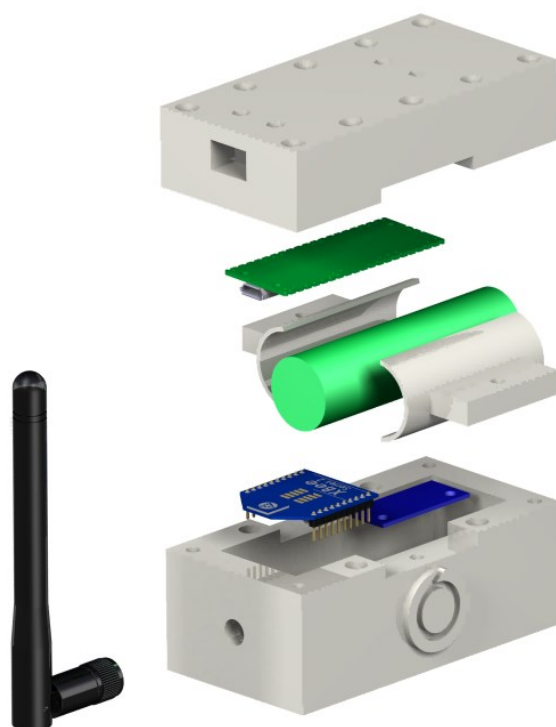


شکل ۲-۲۰: شماتیک اتصالات الکترونیکی قطعات اجزای بخش قابل حمل

## ۲-۷-۵- طراحی محفظه‌ی دستگاه

پس از اتصال قطعات و اجزا به یک‌دیگر، نیاز به طراحی محفظه‌ای وجود داشت تا این قطعات را دربرگیرد. هدف اصلی این بود که سیستم نمونه‌برداری ارتعاشی در اندازه‌ی بسیار کوچک طراحی و توسعه یابد. طرح‌ها و ایده‌های متفاوتی بررسی شد و در نهایت تصمیم گرفته شد تا این محفظه به صورت دو تکه که از وسط از یکدیگر جدا می‌شوند، طراحی شود. این طراحی اجازه می‌دهد تا ماژول‌ها و برد توسعه همگی به بدنه پیچ شوند و باتری نیز در جای خود محکم شود. همچنین فضای خالی به کمترین حد رسیده و در نهایت این دستگاه در کوچکترین ابعاد طراحی شود. همچنین برای اتصال برد توسعه به رایانه (در صورت لزوم) توسط کابل Micro USB، اتصال آنتن به ماژول فرستنده و نیز دکمه‌ی روشن/خاموش کردن دستگاه، فضای لازم در نظر گرفته شد.

مدل‌سازی این بدنه و جانمایی نهایی آن‌ها با استفاده از نرم‌افزار Catia V5 صورت گرفت. نقشه‌ی انفجاری این بدنه را در شکل ۲-۲۱ مشاهده می‌کنید.



شکل ۲- ۲۱: نقشه‌ی انفجاری بدنه‌ی دستگاه

ماژول فرستنده و ماژول شتاب‌سنج در بخش پایینی بدنه قرار می‌گیرند، آنتن به رویه‌ی جلویی دستگاه و دکمه‌ی روشن/ خاموش نیز به بخش پشتی بدنه متصل می‌شود. باتری توسط دو قطعه در وسط دستگاه ثابت می‌شود و در نهایت در بخش بالایی، برد توسعه قرار دارد که محفظه‌ای جهت اتصال آن به رایانه تعبیه شده‌است. این دو قطعه در نهایت توسط شش پیچ M3 به یکدیگر پیچ می‌شوند.

برای اتصال خارجی بدنه‌ی دستگاه به پمپ و لوله‌ها، یک آهنربای نئودیم به ابعاد  $20 \times 20 \times 2$  میلی‌متری در داخل بخش پایینی بدنه جای‌گذاری شده‌است. همچنین برای اتصال بدنه به بست خارجی (در صورت وجود)، چهار سوراخ مناسب برای پیچ M5 پیش‌بینی شده‌است.

این بدنه با استفاده از پرینت سه بعدی (فیلامنت از جنس PLA با قطر ۲۰۰ میکرون در رنگ سفید) ساخته شد و در نهایت پس از اتصال همه‌ی بخش‌های این قطعه به یکدیگر، محصول نهایی به شکل ۲- ۲۲ در آمد.



شکل ۲-۲۲: مدل نهایی پرینت شده‌ی بدنه‌ی دستگاه

پس از آماده‌سازی نهایی این بخش قابل حمل از مجموعه‌ی تشخیص خطا، به سراغ بخش پردازش مرکزی مجموعه می‌رویم.

## ۲-۸- سخت افزار بخش پردازش مرکزی

### ۲-۸-۱- توضیحات کلی

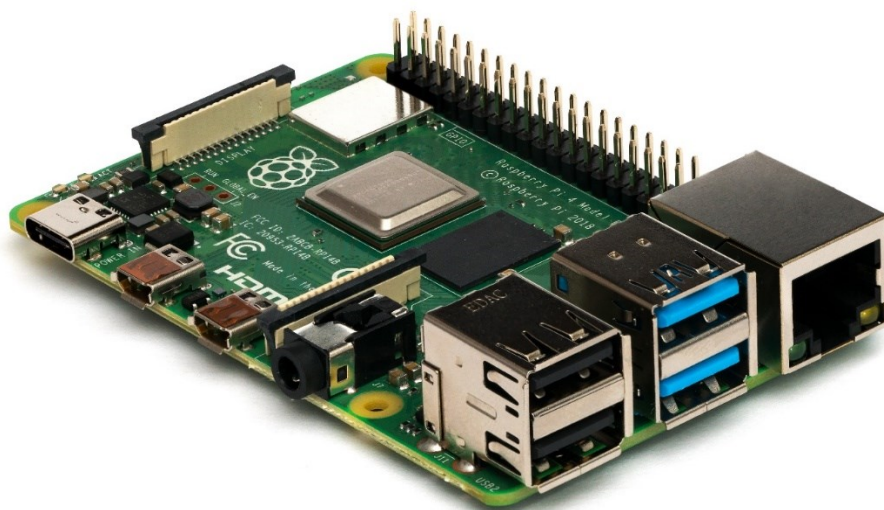
همان‌طور که مطرح شد، پردازش اصلی در بخشی دیگر به جز قطعه‌ی مربوط به داده‌برداری انجام می‌گیرد. مجموعه‌ی عملیات‌هایی که باید در این بخش صورت بگیرد، نیازمند توان پردازشی بیشتری است. هم‌چنین در یک مجموعه‌ی صنعتی واقعی، معمولاً برای هر دستگاه یک بخش پردازش مرکزی وجود دارد و به تعداد ماشین‌آلات صنعتی که باید پایش شوند، دستگاه داده‌برداری مورد نیاز است. بنابراین، این بخش مرکزی باید قادر باشد اطلاعات را از طریق مازول گیرنده‌ی بی‌سیم دریافت کرده و با یک مدل شبکه‌ی عصبی که از قبل آموزش دیده‌است، پیش‌بینی کند هر نمونه نشان‌دهنده‌ی کدام حالت سیستم است. هم‌چنین این داده‌ها باید در یک دیتابیس ذخیره شود تا تاریخچه‌ی کاری دستگاه قابل دسترس باشد. در نهایت برای نمایش بهتر نتایج به کاربر، باید یک رابط کاربری گرافیکی نیز در نظر گرفته شود تا این زنجیره‌ی تشخیص خطا به بهترین شکل تشکیل شود.

باتوجه به مواردی که در بخش قبل بحث شد، سخت افزار بخش مرکزی سیستم عیب‌یابی باید شامل یک مازول گیرنده‌ی بی‌سیم (مانند بخش فرستنده، از نوع XB24-Z7UIT-004 باشد). دیگر موارد لازم نیز شامل یک صفحه نمایش برای نمایش اطلاعات و یک کامپیوتر است. برای صفحه نمایش از یک مانیتور موجود در آزمایشگاه استفاده

شده است. اما برای کامپیوتر، به جای استفاده از کامپیوتر رومیزی، تصمیم بر آن شد تا از کامپیوتری تا حد امکان ارزان قیمت استفاده شود.

## ۲-۸-۲- انتخاب کامپیوتر

معیارهای انتخاب یک کامپیوتر برای پردازش مرکزی سیستم تشخیص عیب شامل مواردی چون قیمت بسیار پایین، پشتیبانی از پورتهای لازم، پشتیبانی از صفحه‌ی نمایشگر و موس و کیبورد، پشتیبانی از سیستم‌عامل دارای رابط کاربری گرافیکی و در نهایت ابعاد کوچک و قابل حمل است. در میان گزینه‌های موجود در بازار ایران، کامپیوترهای Raspberry Pi این قابلیت‌ها را به بهترین شکل ارائه می‌دهند. همچنین موجود بودن این کامپیوتر در آزمایشگاه و کدهای لازم برای بخش داشبورد گرافیکی و دریافت داده‌ها از مازول بی‌سیم، از دیگر دلایل اصلی انتخاب رزبری بود. تصویر این کامپیوتر در شکل ۲-۲۳ آمده است.



شکل ۲-۲۳: تصویر کامپیوتر Raspberry Pi 4

رزبری پای رایانه کوچکی است که از سال ۲۰۰۶ در حال توسعه است و قطعات آن روی یک مادربرد به اندازه کارت بانکی سوار شده و Raspbian را اجرا می‌کند که یک نسخه اختصاصی از سیستم عامل لینوکس است که اختصاصاً برای این رایانه طراحی شده است. [25]

در این پروژه، از Raspberry Pi 4 استفاده شده است که آخرین نسخه‌ی ارائه شده از این سری است. برخی از مشخصات این کامپیوتر در جدول ۲-۴ ارائه شده است.

جدول ۲-۴: مشخصات کامپیوتر Raspberry Pi 4 [26]

Name	Value
SoC	Broadcom BCM2711, Quad coreCortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
GPU	Broadcom Video Core VI
RAM	8GB LPDDR4-3200 SDRAM
Bluetooth	Bluetooth 5.0, BLE
Display port	2 × micro-HDMI 2.0
USB	2x USB 3.0 + 2x USB 2.0
Power Supply	5V via USB type-C up to 3A and GPIO header up to 3A
Wifi	2.4 GHz and 5 GHz 802.11b/g/n/ac wireless LAN
Storage	microSD card
Ethernet	Native Gigabit Ethernet

لازم به ذکر است که این کامپیوتر دارای ورژن‌های قدیمی‌تر و ارزان‌تری نیز هست، اما به دلیل موجود بودن این نسخه، از همان استفاده شد.

در نهایت پس از انتخاب قطعات سخت افزار، به بحث نرم‌افزاری موجود می‌پردازیم.

## ۲-۹- نرم‌افزار بخش پردازش مرکزی

انتخاب موارد نرم‌افزاری عمدتاً بر اساس نرم‌افزارها و کدهای موجود و آماده در اینترنت و نیز پروژه‌های قبلی انجام‌شده در آزمایشگاه تبدیل انرژی هوشمند بوده است. برای سیستم عامل کامپیوتر که رابط کاربری گرافیکی

نیز داشته باشد از Raspberry Pi OS with Desktop، برای دیتابیس از InfluxDB و برای داشبورد گرافیکی از Grafana استفاده شده است.

InfluxDB یک پایگاه داده سری زمانی منبع باز است که توسط شرکت InfluxData توسعه یافته است. این دیتابیس به زبان برنامه نویسی Go نوشته شده است و برای ذخیره و بازیابی داده های سری زمانی در زمینه هایی مانند پایش سیستم ها، داده های حسگر اینترنت اشیا و تجزیه و تحلیل داده های برخط مورد استفاده قرار می گیرد.[27]

گرافانا نرم افزاری چند پلتفرم و منبع باز برای تجزیه و تحلیل و بصری سازی تحت وب است که هنگام اتصال به منابع داده پشتیبانی شده، می تواند گراف ها، چارت ها و هشدارها را برای وب ارائه دهد. گرافانا به عنوان یک ابزار بصری سازی، جزئی محبوب در پایش است که اغلب در ترکیب با پایگاه داده های سری زمانی مانند InfluxDB استفاده می شود.[28]

\*زنجیره ی دریافت اطلاعات از ماژول بی سیم و انتقال و ذخیره سازی آن در دیتابیس InfluxDB و نیز نمایش داده های موجود در داشبورد گرافانا از قبل تکمیل شده بوده است و در این پروژه، تنها تغییراتی در کدها داده شده است تا داده های دریافتی پیش پردازش شوند و سپس به عنوان ورودی به مدل ذخیره شده ی هوش مصنوعی داده شوند و خروجی آن ها در داشبورد نمایش داده شود.



## نتایج

در این بخش، به بررسی نتایج حاصل از آموزش شبکه‌ی عصبی با استفاده از داده‌های ورودی پرداخته شده است. ابتدا نتایج بررسی با استفاده از فرایند K-fold و سپس ماتریس درهم‌ریختگی آورده شده اند. در انتها نیز برای نمایش بهتر چگونگی فرایند تشخیص عیب در شبکه‌ی عصبی، خروجی‌های هر لایه‌ی آن به تصویر کشیده شده است.

### ۳-۱- نتایج فرایند K-fold

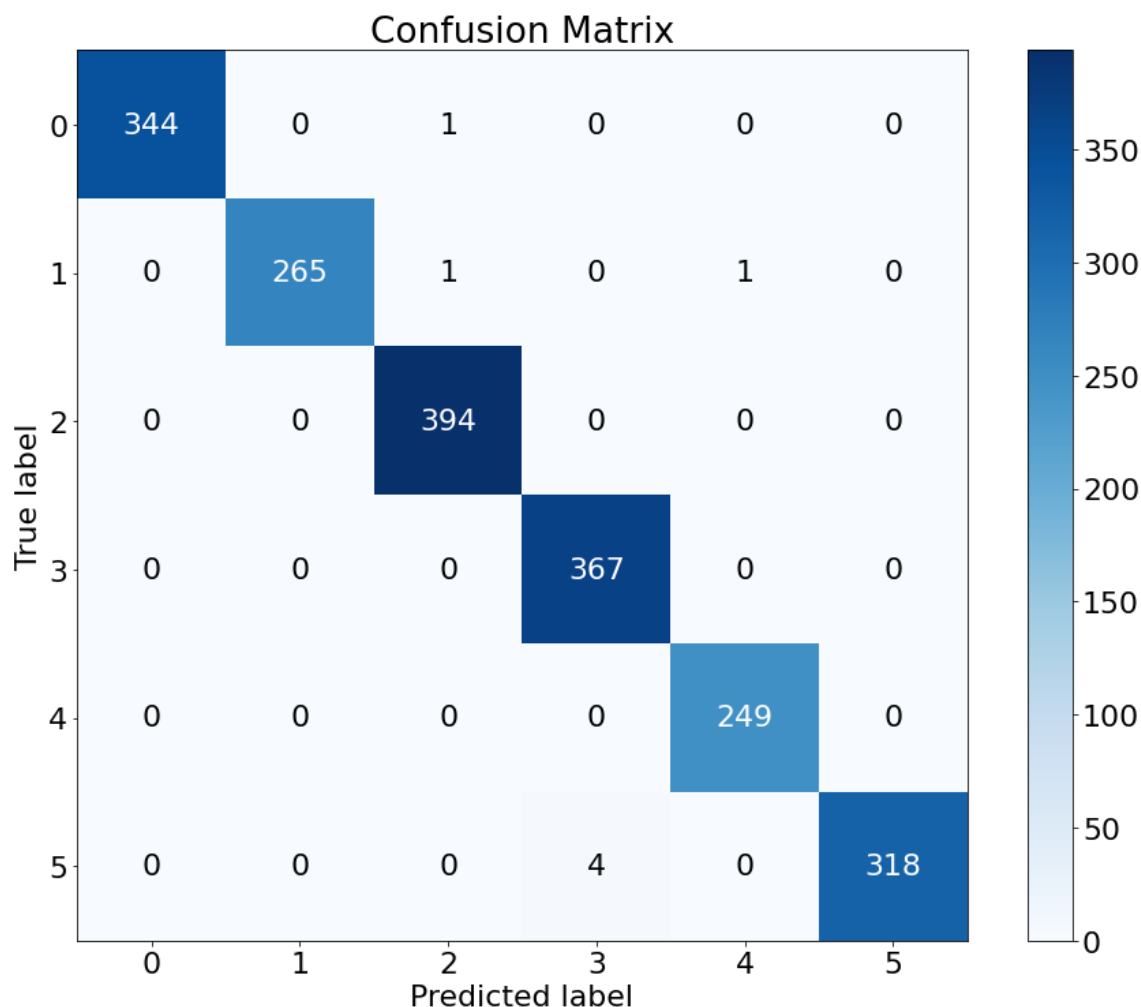
در نهایت پس از کامل شدن آموزش مدل و ارزیابی، نتایج فرایند K-fold به صورت زیر بدست آمده‌است:

جدول ۳-۱: نتایج اعتبارسنجی مدل توسط فرایند K-fold

شماره‌ی fold	دقت (درصد)
۱	۸۵.۴۹
۲	۱۰۰
۳	۱۰۰
۴	۱۰۰
۵	۱۰۰
۶	۹۹.۹
۷	۸۱.۴۸
۸	۹۹.۷
۹	۱۰۰
۱۰	۹۹.۷۹
میانگین	$۹۶.۶۳ \pm ۶.۳۵$

### ۳-۲- ماتریس درهم‌ریختگی

همچنین ماتریس در هم ریختگی<sup>۲۵</sup> داده‌های تست که به عنوان ورودی به شبکه داده شده‌اند، به شکل زیر است. در محور عمودی این ماتریس، برچسب واقعی داده‌ها آمده است و در محور افقی آن نیز برچسب پیش‌بینی شده توسط شبکه‌ی عصبی قرار دارد. در خانه‌های این ماتریس هم تعداد داده‌هایی که در هر دسته قرار گرفته‌اند، نمایش داده شده‌است.



شکل ۳-۱: ماتریس در هم ریختگی داده‌های تست

همان طور که مشاهده می‌شود، مدل نهایی که با تعداد گام‌های بیشتری آموزش دیده شده‌است، پیش‌بینی وضعیت سلامت و نوع عیب پمپ با دقت بسیار بالایی انجام شده‌است.

### ۳-۳- بصری سازی خروجی لایه‌ها

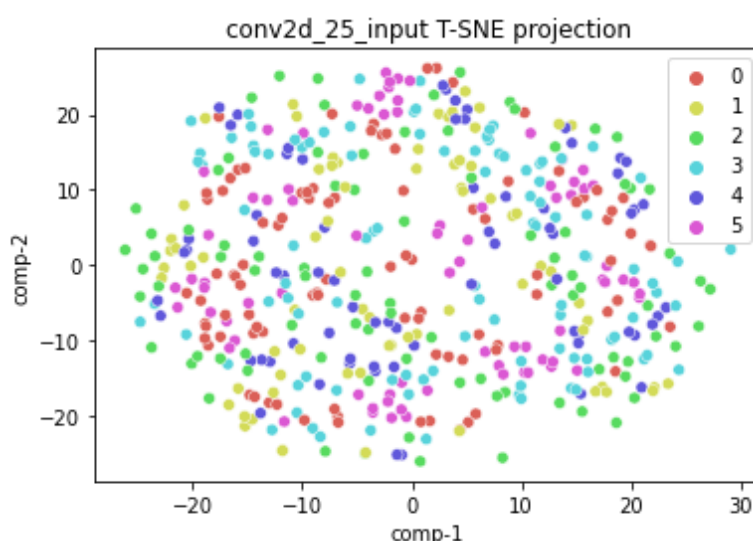
به منظور تحلیل بیشتر نتایج و فرایند طی شده در لایه‌های مختلف شبکه‌ی عصبی، در این بخش به بصری سازی خروجی هر مرحله پرداخته‌ایم. همان طور که در ابتدای این بخش و در تصویر مربوط به معماری شبکه نشان داده شده‌است، خروجی هر لایه از ۹۶۱ در خروجی لایه‌ی اول شروع شده و تا ۶۴ عدد در آخرین لایه پیش از لایه‌ی soft-max، متغیر است. اما برای نمایش یک داده در صفحه، باید از الگوریتمی استفاده کنیم که بتواند این تعداد ویژگی‌های یک داده را به دو عدد کاهش داده و این دو بُعد را طوری انتخاب کند که ویژگی‌های اصلی و وجه

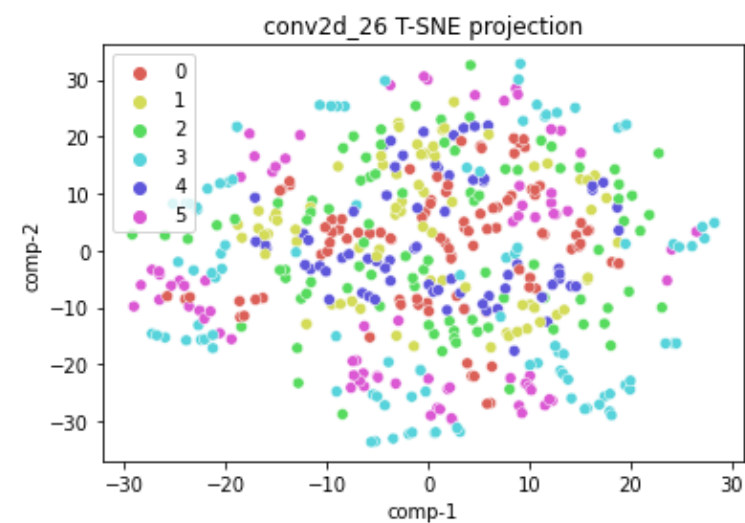
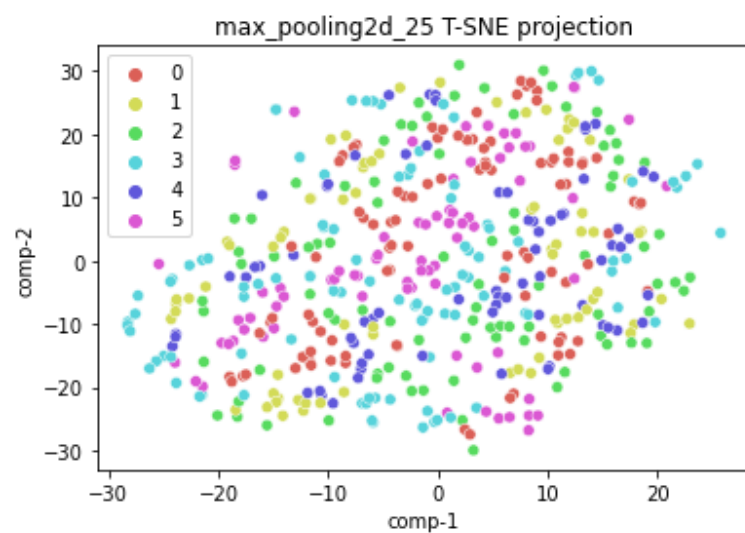
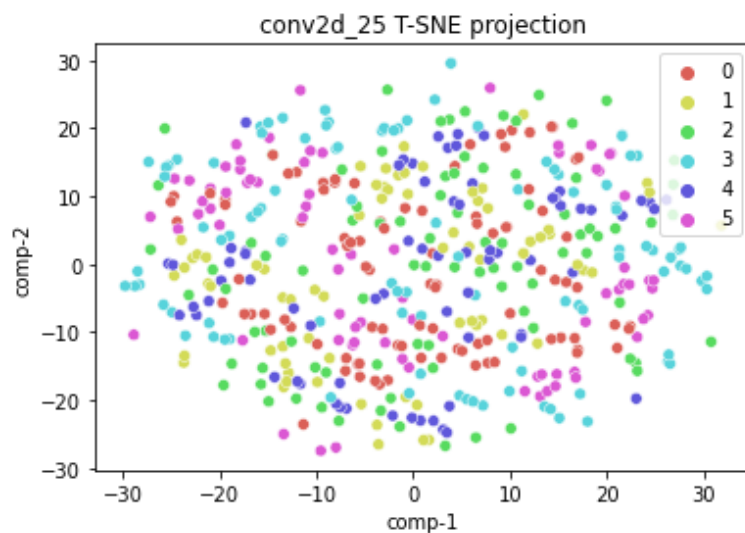
تمایز داده‌ها حفظ شود. الگوریتم‌های مختلفی برای این کار توسعه داده شده‌اند که معروف‌ترین آن‌ها که در مقالات بسیاری برای بصری سازی خروجی لایه‌های شبکه‌ی عصبی به کار رفته است، الگوریتم t-SNE نام دارد.

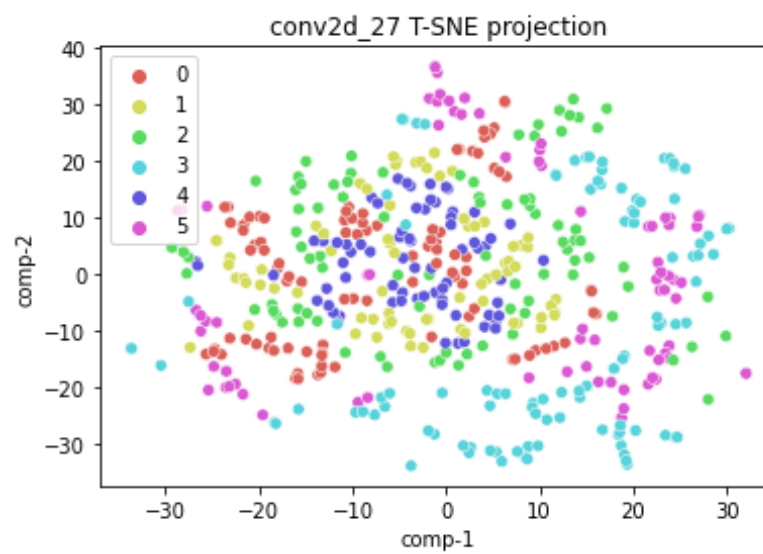
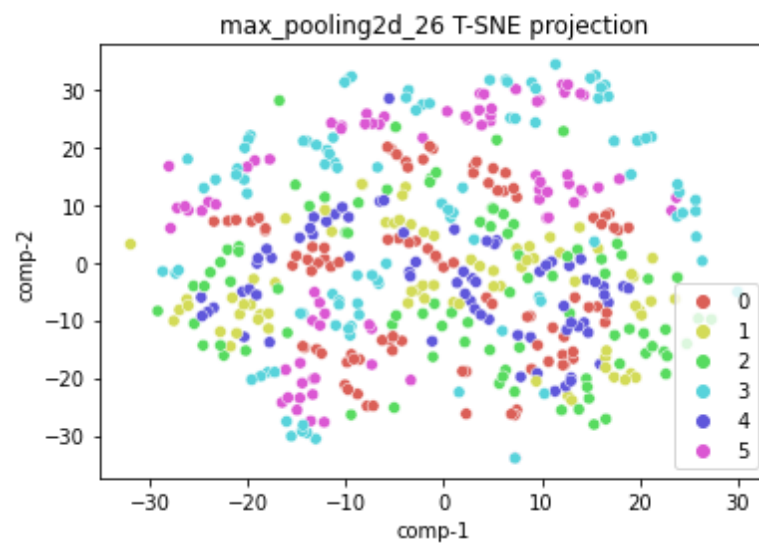
الگوریتم t-SNE یک روش نظارت نشده‌ی غیر خطی است که برای اکتشاف و بصری سازی داده‌ها مورد استفاده قرار می‌گیرد. به بیان ساده‌تر، t-SNE به کاربر درکی از اینکه داده‌ها چگونه در فضای ابعاد بالا سازمان‌دهی شده‌اند ارائه می‌کند. این روش توسط (Laurens van der Maatens) و (Geoffrey Hinton) در سال ۲۰۰۸ ساخته شد. [29]

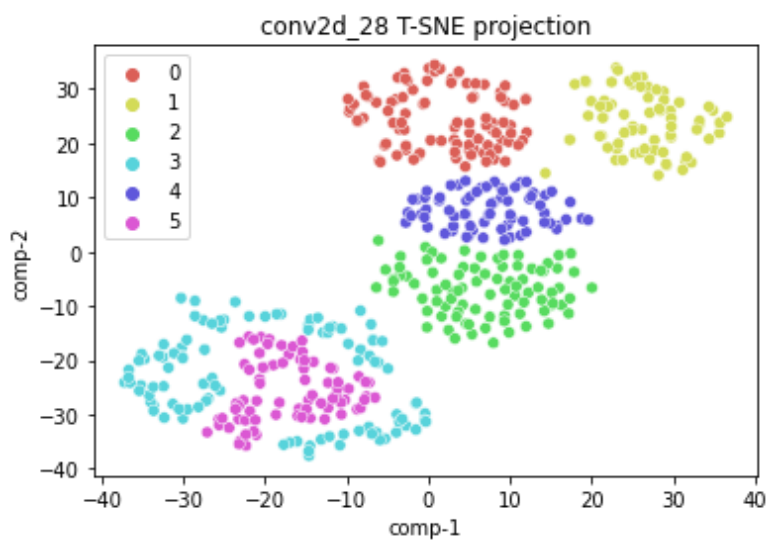
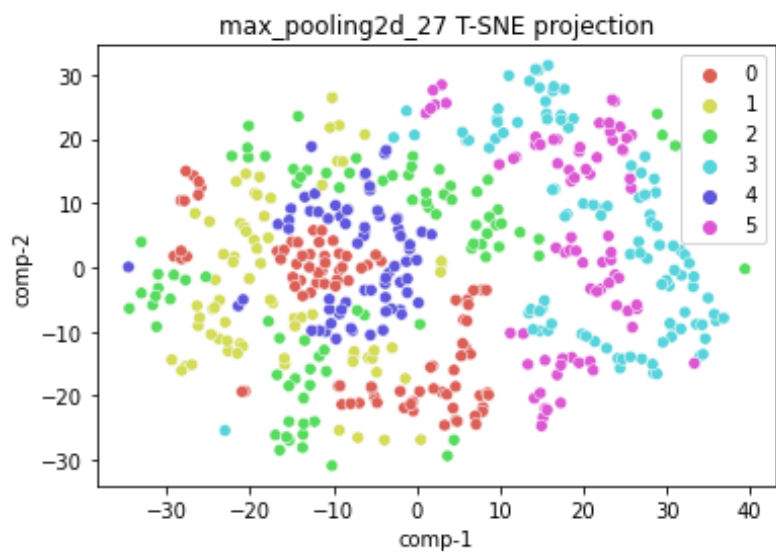
این الگوریتم یک معیار مشابهت را بین جفت نمونه‌ها در داده‌های ابعاد بالا و در فضای ابعاد کم محاسبه و سپس برای بهینه‌سازی این دو معیار مشابهت با استفاده از یک تابع هزینه تلاش می‌کند [30].

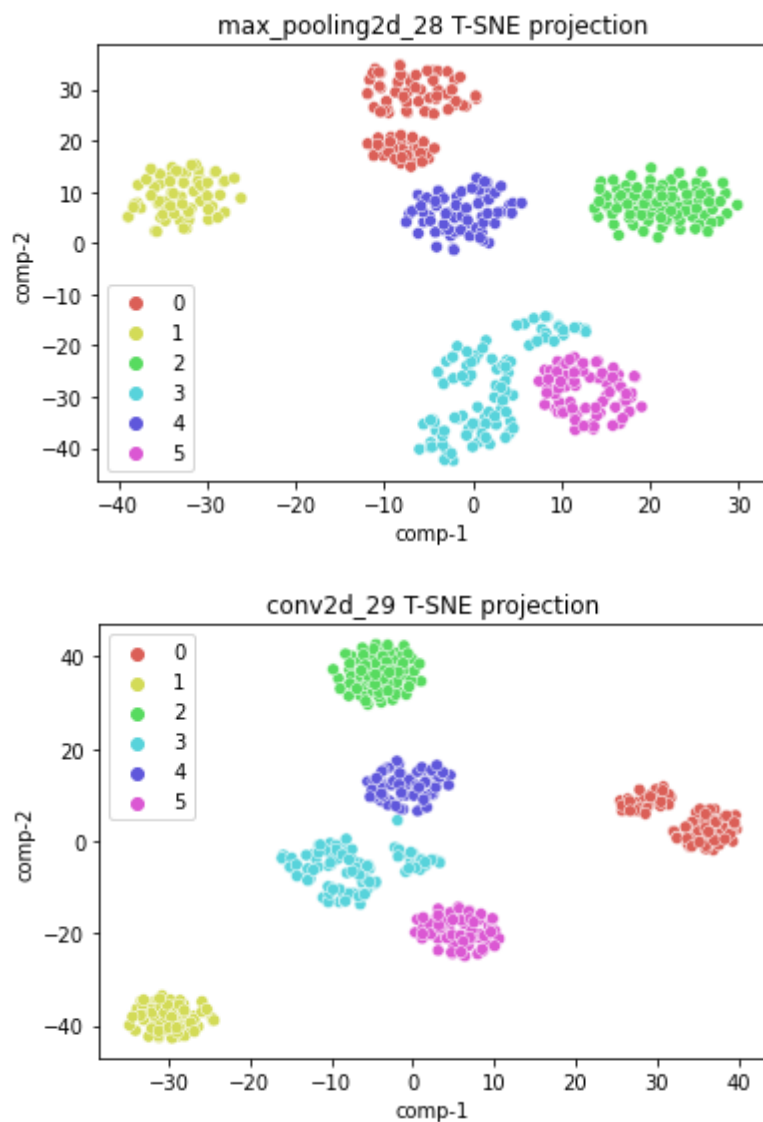
با استفاده از کتابخانه‌ی Keract، خروجی‌های هر لایه از شبکه‌ی عصبی برای هر داده بدست آمده و با کمک t-SNE، در فضای دو بعدی به یک نقطه تبدیل می‌شود. این عمل بر روی ۵۰۰ نقطه‌ی رندوم از برجسب‌های مختلف انجام گرفته‌است و در شکل زیر، به صورت مرحله به مرحله می‌توان مشاهده کرد داده‌های ورودی مربوط به وضعیت‌های مختلف سلامت سیستم که در ابتدا قابل تمایز از یکدیگر نیستند، چگونه پس از عبور از فیلترهای مختلف، به صورت خوشه‌هایی کاملاً جدا از یکدیگر در می‌آیند.



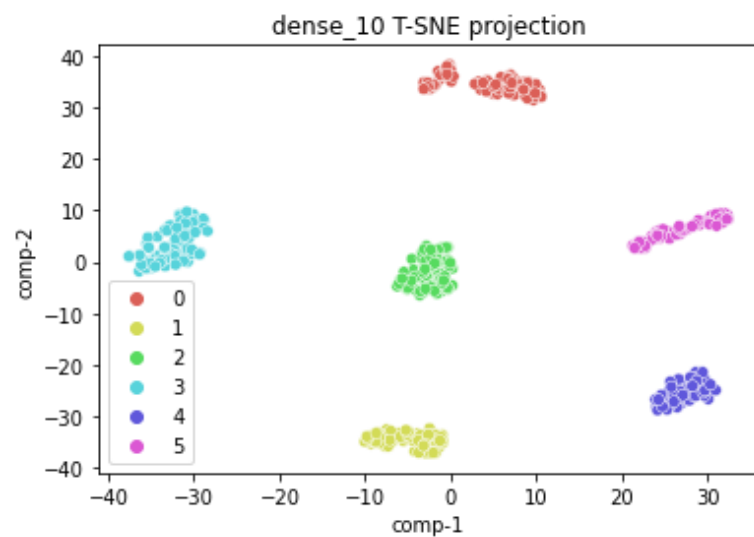
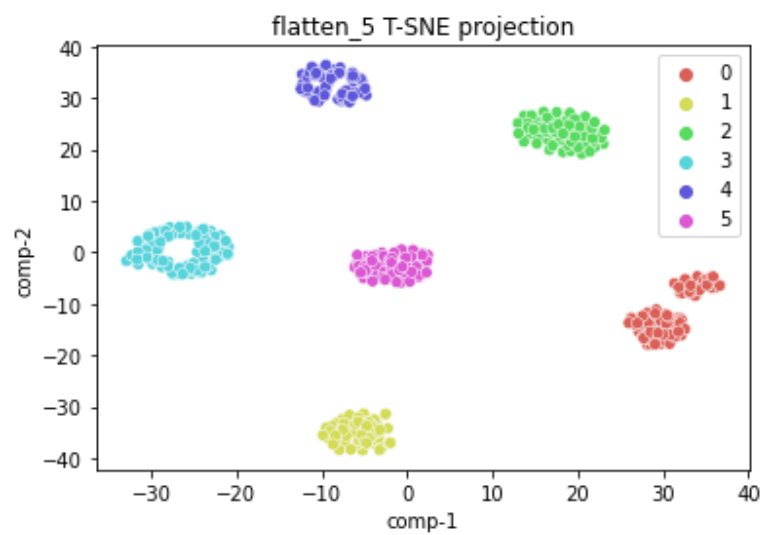
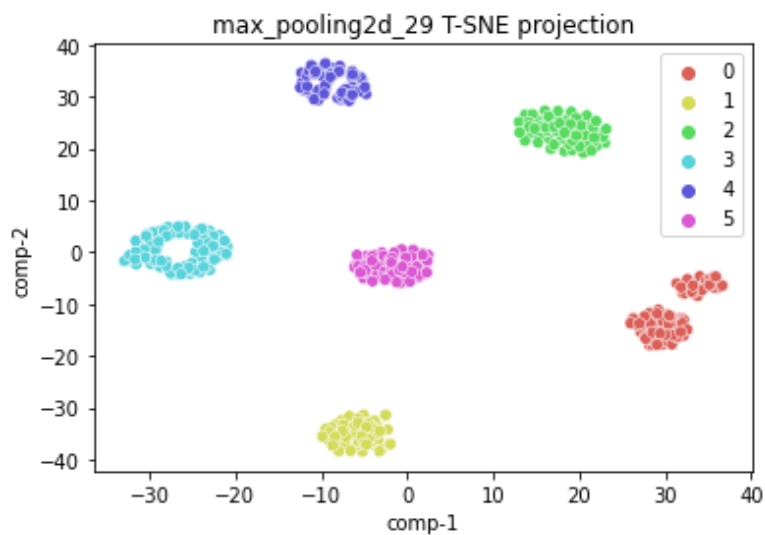


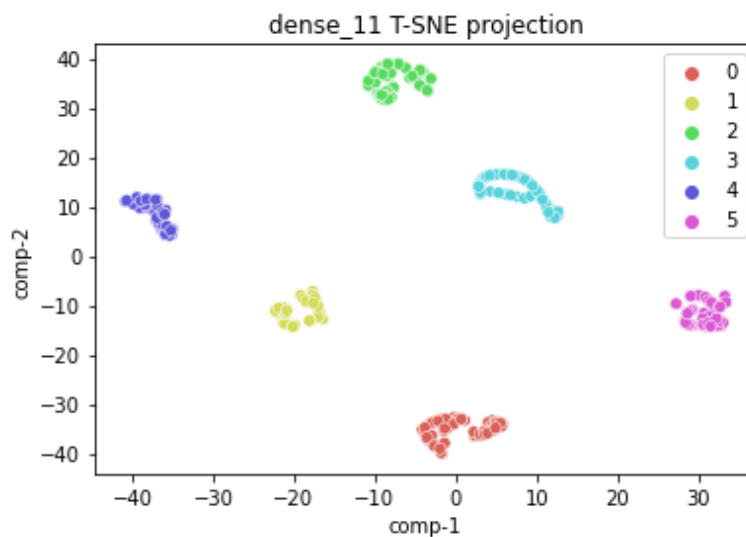












شکل ۳-۲: بصری سازی فرایند جداسازی ۵۰۰ داده‌ی متعلق به دسته‌های متفاوت در لایه‌های شبکه‌ی عصبی

همان طور که در تصاویر بالا نشان داده شده‌است، فرایند جداسازی داده‌ها به خوبی و با دقت بسیار بالا انجام می‌گیرد و با وجود آنکه عیب‌هایی مانند ترک و یا پریدگی نوک پره در سائز کوچک عیب‌هایی بسیار ریز هستند، اما به صورت کامل از دسته داده‌های سالم متمایز شده‌اند و می‌توانند تشخیص داده شوند.

## مراجع

- [1] A. Kupervas, "Predictive Maintenance: What is the Economic Value?" <https://www.anodot.com/blog/predictive-maintenance/> (accessed Sep. 17, 2021).
- [2] "How Manufacturers Achieve Top Quartile Performance - Emerson - Unlocking Top Performance." <https://partners.wsj.com/emerson/unlocking-performance/how-manufacturers-can-achieve-top-quartile-performance/> (accessed Sep. 17, 2021).
- [3] S. Vemuri, "Why Is Predictive Maintenance Important?" <https://www.digitaldoughnut.com/articles/2018/january/why-is-predictive-maintenance-important>.
- [4] "The Benefits of Predictive Maintenance | Shell Lubricant Solutions | Shell Global." <https://www.shell.com/business-customers/lubricants-for-business/perspectives/the-benefits-of-predictive-maintenance.html#vanity-aHR0cHM6Ly93d3cuc2h1bGwuY29tL2J1c2luZXNzLWN1c3RvbWVycy9sdWJyaWNhbnRzLWZvci1idXNpbmVzcy9pbmR1c3RyeS1pbmNpZ2h0cy90aGUtYmVuZWZpdHMtb2YtcHJlZGljdGl2ZS1tYWludGVuYW5jZS5odG1s> (accessed Sep. 17, 2021).
- [5] W. Zhang, X. Li, X. D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Meas. J. Int. Meas. Confed.*, vol. 152, 2020, doi: 10.1016/j.measurement.2019.107377.
- [6] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors (Switzerland)*, vol. 17, no. 2, 2017, doi: 10.3390/s17020425.
- [7] M. Sat, V. S. Rao, and C. G. Devy, "Cepstrum Analysis-An Advanced Technique in Vibration Analysis of Defects in Rotating Machinery," *Def. Sci. J.*, vol. 44, no. 1, pp. 53–60, 1994.
- [8] P. K. Kankar, S. C. Sharma, and S. P. Harsha, "Fault diagnosis of ball bearings using continuous wavelet transform," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2300–2312, Mar. 2011, doi: 10.1016/J.ASOC.2010.08.011.
- [9] Q. Hu, Z. He, Z. Zhang, and Y. Zi, "Fault diagnosis of rotating machinery based on improved wavelet package transform and SVMs ensemble," *Mech. Syst. Signal Process.*, vol. 21, no. 2, pp. 688–705, Feb. 2007, doi: 10.1016/J.YMSSP.2006.01.007.
- [10] X. D. T. Han, B.-S. Yang, and S.-J. Lee, "Application of Random Forest Algorithm in Machine Fault Diagnosis," *Proc. 1st World Congr. Eng. Asset Manag. WCEAM 2006*, pp. 779–784, 2006, doi: 10.1007/978-1-84628-814-2\_82.
- [11] R. Huang, Y. Liao, S. Zhang, and W. Li, "Deep Decoupling Convolutional Neural Network for Intelligent Compound Fault Diagnosis," *IEEE Access*, vol. 7, no. 1, pp. 1848–1858, 2019, doi: 10.1109/ACCESS.2018.2886343.
- [12] X. Li, W. Zhang, Q. Ding, and J.-Q. Sun, "Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation," *J. Intell. Manuf.*, vol. 31, no. 2, pp. 433–452, 2020, doi: 10.1007/s10845-018-1456-1.

- 
- 
- [13] J. Sun, C. Yan, and J. Wen, "Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 1, pp. 185–195, 2018, doi: 10.1109/TIM.2017.2759418.
- [14] "TensorFlow," [Online]. Available: <https://en.wikipedia.org/wiki/TensorFlow>.
- [15] "Scikit-learn," [Online]. Available: <https://en.wikipedia.org/wiki/Scikit-learn>.
- [16] P. Remy, "Keract: A library for visualizing activations and gradients." <https://github.com/philipperemy/keract> (accessed Sep. 17, 2021).
- [17] A. Reybod, "Cross-Validation," [Online]. Available: <https://blog.faradars.org/cross-validation/>.
- [18] E. Hesarakhi, "DBSCAN Algorithm." <https://blog.faradars.org/dbscan-algorithm-in-python/>.
- [19] A. Reybod, "K-means Clustering Algorithm." <https://blog.faradars.org/k-means-clustering-algorithm/>.
- [20] WikiPedia, "Accelerometer." <https://en.wikipedia.org/wiki/Accelerometer>.
- [21] WikiPedia, "ZigBee." <https://en.wikipedia.org/wiki/Zigbee>.
- [22] DigiKey, "XBee Module." <https://www.digikey.com/en/products/detail/digi/XB24-Z7UIT-004/1942303>.
- [23] K. Hart, "Pi Pico." <https://all3dp.com/2/raspberry-pi-pico-review-specs/>.
- [24] "MicroPython." <https://micropython.org/>.
- [25] M. Lotfi, "RaspBerry Pi." <https://blog.faradars.org/رزبری-پای-چیست/>.
- [26] "Raspberry Pi 4 Tech Specs." <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>.
- [27] J. Turnbull, *The Art of Monitoring*. James Turnbull, 2014.
- [28] WikiPedia, "Grafana." [https://en.wikipedia.org/wiki/Grafana#cite\\_note-enterprise-2](https://en.wikipedia.org/wiki/Grafana#cite_note-enterprise-2).
- [29] L. Van Der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. February, pp. 2579–2625, 2008.
- [30] E. Hesarakhi, "t-SNE Method." <https://blog.faradars.org/t-sne-مقدمه-ای-بر-الگوریتم/>.



University of Tehran  
College of Engineering  
School of Mechanical Engineering

# **Centrifugal pump health monitoring using Neural networks**

A thesis submitted to the School of Mechanical Engineering  
In partial fulfillment of the requirements for  
The degree of BSc in  
Mechanical Engineering

**Mahdi Abdollah Chalaki**

**Supervisor:**

**Dr. Ali Sadighi**

**September 2021**