

KNO-1204-5004

## تشخیص خطا در حسگرهای اثر هال موتور DC بدون جاروبک با استفاده از

### شبکه‌های عصبی و مدل یادگیری عمیق

<sup>۱</sup> moslem.494.63@gmail.com مسلم رحیم پور

<sup>۲</sup> m.moghadasian@scu.ac.ir محمود مقدسیان

<sup>۱</sup> کارشناسی ارشد پردیس صنعتی شهدای هویزه- دانشگاه شهید چمران اهواز-سوسنگرد- ایران

<sup>۲</sup> استادیار پردیس صنعتی شهدای هویزه- دانشگاه شهید چمران اهواز-سوسنگرد- ایران

**چکیده:** این پژوهش به طبقه‌بندی چندحالتی خطاهای سنسورهای اثر هال در موتورهای BLDC با به‌کارگیری مدل‌های یادگیری عمیق می‌پردازد. مدل‌های مورد بررسی شامل CNN یک‌بعدی (Conv1D)، CNN دوبعدی (Conv2D)، Transformer و یک مدل ترکیبی CNN+LSTM هستند. داده‌های مورد استفاده، جریان‌های ورودی الکتروموتور در دو حالت سالم و خطا دار می‌باشند که برای هر مدل، پیش‌پردازش متناسب با ساختار ورودی آن انجام شده است. هدف نهایی پژوهش، طراحی و ارزیابی مدل ترکیبی CNN+LSTM است که با دستیابی به دقت ۹۳٪ در داده‌های آموزش و ۹۱٪ در داده‌های آزمون، بهترین عملکرد را در میان مدل‌های بررسی شده نشان می‌دهد. در مقایسه، سایر مدل‌ها به دقت‌های پایین‌تری رسیده‌اند. این نتایج حاکی از آن است که انتخاب مدل باید با توجه به هدف مشخص و ویژگی‌های مسئله صورت گیرد. در این راستا، برای دستیابی به نتیجه مطلوب در مدل CNN+LSTM، استفاده از تکنیک‌های مهندسی ویژگی ضروری تشخیص داده شد.

**کلید واژه‌ها:** دسته بندی سری های زمانی، شبکه های عصبی، مدل یادگیری عمیق، موتور BLDC

#### ۱. مقدمه

موتورهای جریان مستقیم بدون جاروبک (Brushless DC Motors - BLDC) بطور گسترده‌ای در صنایع مختلف و سامانه‌های صنعتی پیشرفته مورد استفاده قرار گرفته‌اند. یکی از اجزای کلیدی در عملکرد صحیح این موتورها، سیستم تشخیص موقعیت روتور است که معمولاً با استفاده از حسگرهای اثر هال پیاده‌سازی می‌شود [۱]. حسگرهای اثر هال با تشخیص موقعیت زاویه‌ای روتور، اطلاعات لازم برای کموتاسیون الکترونیکی جریان در سیم‌پیچ‌های استاتور را فراهم می‌کنند. بروز هرگونه خطا در این حسگرها می‌تواند موجب افت عملکرد یا از کار افتادن موتور شود [۱ و ۲].

یادگیری ماشین به‌عنوان شاخه‌ای از هوش مصنوعی، به مطالعه الگوریتم‌هایی می‌پردازد که با استفاده از داده‌های تجربی (نمونه‌های ورودی و خروجی متناظر) یک نگاشت تقریبی از فضای ورودی به فضای هدف را استنتاج می‌کنند. در این فرایند، مدل با مشاهده حجم انبوهی از نمونه‌های آموزشی، تابعی را می‌آموزد که بتواند ورودی‌های جدید را به خروجی‌های صحیح نگاشت کند [۳-۵].

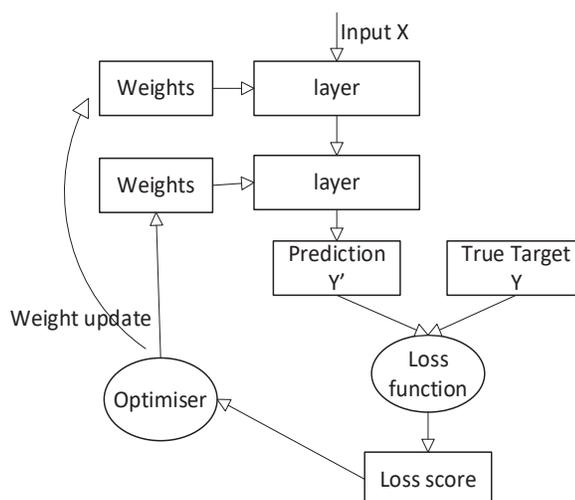
شبکه‌های عصبی عمیق، این نگاشت را از طریق یک سلسله‌مراتب چندلایه از تبدیل‌های غیرخطی پارامتری شده محقق می‌سازند. هر لایه متشکل از یک مجموعه وزن (پارامتر) است که نحوه تبدیل داده‌های ورودی آن لایه را تعیین می‌کند. از دیدگاه ریاضی، هر لایه یک

تابع  $f_i(\mathbf{x}; \mathbf{W}_i)$  است که در آن ماتریس وزن‌های لایه  $i$ ام بوده و خروجی لایه قبلی ( $\mathbf{x}$ ) را به فضای ویژگی‌های انتزاعی‌تر تبدیل می‌کند.

فرایند یادگیری در این شبکه‌ها معادل بهینه‌سازی پارامترهای تمام لایه‌هاست، به گونه‌ای که تابع خطای کل (Loss Function) بین خروجی شبکه و برچسب‌های واقعی کمینه شود. در شبکه‌های عمیق مدرن، تعداد این پارامترها ممکن است به ده‌ها یا صدها میلیون برسد؛ بنابراین مسئله یافتن مقادیر بهینه در فضایی با ابعاد بسیار بالا صورت می‌گیرد [۶-۹].

کلید حل این مسئله در استفاده از روش‌های بهینه‌سازی مبتنی بر گرادیان است. در این چارچوب، تابع خطا ( $\mathcal{L}$ ) به عنوان معیار سنجش فاصله خروجی شبکه از هدف عمل می‌کند. با محاسبه گرادیان تابع خطا نسبت به هر پارامتر ( $\nabla_{\mathbf{W}_i} \mathcal{L}$ )، الگوریتم بهینه‌ساز (نظیر SGD، Adam و ...) جهت به‌روزرسانی وزن‌ها در خلاف جهت گرادیان (یعنی جهت کاهش خطا) را تعیین می‌کند. این فرایند تکراری که تحت عنوان حلقه آموزش شناخته می‌شود، در هر تکرار (epoch) بر روی دسته‌ای از نمونه‌ها (batch) اجرا شده و پارامترها را گام‌به‌گام به سمت نقطه کمینه محلی سوق می‌دهد [۱۰-۱۳].

مقداردهی اولیه وزن‌ها معمولاً به صورت تصادفی و مطابق با توزیع‌های خاص (نظیر He یا Xavier) انجام می‌گیرد. خروجی شبکه فاصله قابل توجهی با مقادیر هدف دارد و تابع خطا مقدار بزرگی را نشان می‌دهد. با پیشروی در فرایند آموزش و انجام به‌روزرسانی‌های متعدد، شبکه به تدریج هم‌گرا شده و خطای آموزش کاهش می‌یابد. شبکه‌ای که پس از اتمام آموزش به حداقل خطای ممکن (در داده‌های آموزشی و با در نظرگیری مکانیزم‌های جلوگیری از بیش‌برازش) دست یافته باشد، شبکه آموزش‌دیده نامیده می‌شود. شکل ۱ نمای شماتیک از این فرایند کلی را نمایش می‌دهد [۴].



شکل ۱. دیاگرام کلی عملکرد مدل‌های یادگیری عمیق

## ۲. روش انجام کار و ساختار مجموعه داده

برای پیش پردازش داده‌ها و همه مدل‌های طراحی شده مبتنی بر یادگیری عمیق از بستر google colab استفاده شده است. اضافه بر اینکه این بستر همه کتابخانه‌های مورد نیاز برای کدنویسی پایتون را در خود دارد، یک حافظه و گرافیک قابل قبول رایگان در اختیار کاربر قرار می‌دهد. به دلایل ذکر شده استفاده از google colab برای نوشتن کدهای پایتون اجتناب‌ناپذیر است.

برای این منظور، یک مجموعه داده دقیق و ساختاریافته شامل اندازه‌گیری‌های جریان فاز در شرایط مختلف جابه‌جایی حسگر‌ها تهیه شده است. این داده‌ها در چهار دسته بدون تأخیر، تأخیر ۰,۰۰۱ ثانیه، تأخیر ۰,۰۰۵ ثانیه و تأخیر ۰,۰۱ ثانیه طبقه‌بندی شده‌اند. برای آموزش مدل‌های یادگیری عمیق، داده‌ها بر اساس وضعیت عملکرد حسگرهای اثر‌هاال برچسب‌گذاری شده‌اند. به هر نمونه از سیگنال‌های

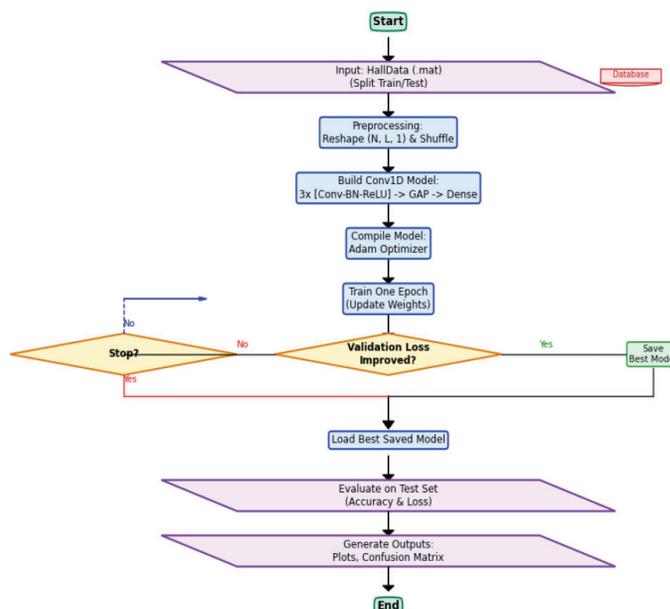
جریان سه‌فاز، یک برجسب عددی اختصاص داده شده که بیانگر نوع حالت عملکرد موتور (سالم یا نوع خاصی از خرابی حسگر هال) است. عدد ۰ بیانگر حالت سالم و اعداد ۱، ۲ و ۳ به ترتیب حالت های خطا با تاخیرهای زمانی ۰،۰۱، ۰،۰۵ و ۰،۰۰۱ ثانیه را در بر می گیرد [۶].

### ۳. روش های مورد استفاده تشخیص و طبقه بندی خطا

پس از انجام پیش پردازش، داده‌ها به مجموعه‌های آموزش و آزمون تقسیم شده و به‌عنوان ورودی به مدل‌های مختلف یادگیری عمیق شامل Conv1D، Conv2D، CNN-LSTM و Transformer اعمال می‌شوند.

#### ۱،۳ ساختار شبکه Conv1D

تعداد کل پارامترهای شبکه در حدود ۱۸ هزار پارامتر بوده که این موضوع، احتمال بیش‌برازش (Overfitting) را کاهش می‌دهد. پس از لایه ورودی، سه بلوک Conv1D با تعداد فیلترهای افزایشی (۱۶، ۳۲ و ۶۴) به کار گرفته شده‌اند. از لایه Global Average Pooling به جای لایه‌های Flatten متداول استفاده شده است که باعث کاهش پیچیدگی مدل و افزایش پایداری آموزش می‌شود. در انتها، یک لایه Dense با تابع فعال‌ساز Relu و یک لایه خروجی Softmax برای طبقه‌بندی چهار کلاس در نظر گرفته شده است. ورودی شبکه به‌صورت یک تنسور با ابعاد (۱ × ۲۰۰۰) تعریف شده است، که در آن بعد دوم نشان‌دهنده یک کانال ورودی است. لایه‌های Conv1D نقش اصلی را در استخراج ویژگی‌های زمانی از سیگنال‌های جریان ایفا می‌کنند. در لایه اول، فیلترهایی با طول کرنل بزرگ (۵۱ نمونه) به کار گرفته شده‌اند تا الگوهای زمانی با مقیاس بزرگ، مانند تغییرات ناشی از کموتاسیون موتور، استخراج شوند. در لایه‌های بعدی، طول کرنل‌ها به ترتیب به ۳۱ و ۲۱ کاهش یافته است تا ویژگی‌های ظریف‌تر و محلی‌تر سیگنال شناسایی شوند. پس از هر لایه کانولوشنی، از لایه‌های MaxPooling1D با ضریب کاهش بالا استفاده شده است که باعث کاهش ابعاد داده و افزایش تمرکز شبکه بر ویژگی‌های غالب سیگنال می‌شود. در انتها، لایه Global Average Pooling میانگین ویژگی‌های استخراج شده را محاسبه کرده و یک نمایش فشرده و معنادار از سیگنال ورودی ایجاد می‌کند. برای درک بهتر ساختار مدل، شکل ۲ ساختار کلی کد پایتون و مدل را بصورت یک فلوجارت نمایش می‌دهد.

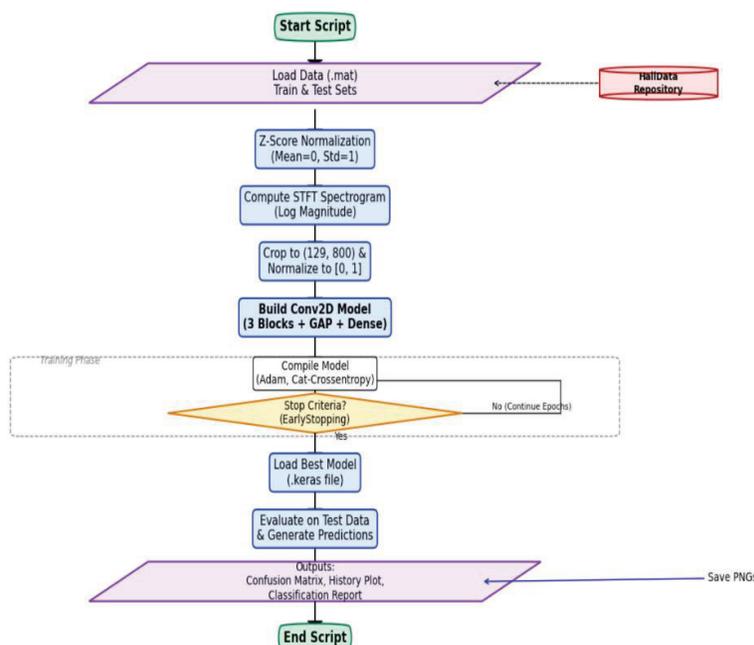


شکل ۲. نمایش ساختار کلی کد پایتون و مدل Conv1D

#### ۲،۳ ساختار مدل Conv2D

در این روش ابتدا سیگنال‌های جریان به حوزه زمان-فرکانس منتقل شده و سپس الگوهای دوبعدی حاصل با استفاده از شبکه Conv2D تحلیل می‌شوند. این رویکرد امکان استخراج هم‌زمان ویژگی‌های زمانی و فرکانسی مرتبط با خطاهای ناشی از تأخیر حسگرهای هال را فراهم می‌کند [۴].

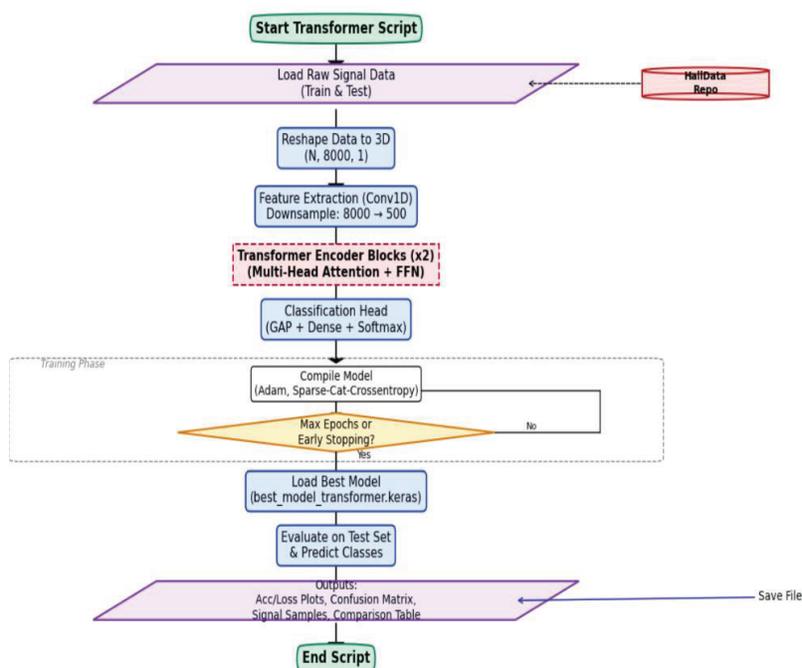
ورودی شبکه یک تانسور چهاربعدی با ابعاد  $(1 \times 800 \times 129)$  است که نشان‌دهنده طیف STFT تک‌کاناله هر نمونه می‌باشد. این شبکه شامل سه لایه Conv2D با تعداد فیلترهای افزایشی  $(32, 64, 96)$  است که در هر لایه، فیلترهایی با ابعاد متفاوت به کار گرفته شده‌اند تا الگوهای محلی در مقیاس‌های مختلف زمانی و فرکانسی استخراج شوند. استفاده از Batch Normalization پس از هر لایه کانولوشنی موجب افزایش سرعت همگرایی و کاهش حساسیت شبکه به مقاردهی اولیه وزن‌ها شده است. فیلترهای دوبعدی شبکه Conv2D به صورت هم‌زمان تغییرات فرکانسی و زمانی موجود در سیگنال‌های جریان را تحلیل می‌کنند. این ویژگی به ویژه برای تشخیص خطاهای مبتنی بر تأخیر زمانی حسگرهای هال اهمیت دارد، زیرا این خطاها نه تنها در حوزه زمان، بلکه در الگوی توزیع انرژی فرکانسی سیگنال نیز اثرگذار هستند. شکل ۳ ساختار کلی مدل کانولوشنال ۲ بعدی را به صورت جدول نمایش می‌دهد. در این ساختار، لایه‌های MaxPooling2D با ضرایب کاهش متفاوت در راستاهای زمان و فرکانس به کار گرفته شده‌اند تا ابعاد داده کاهش یافته و تمرکز شبکه بر الگوهای غالب افزایش یابد. در انتها، لایه Global Average Pooling 2D جایگزین لایه‌های تمام‌متصل حجیم شده و یک بردار ویژگی فشرده و مقاوم در برابر بیش‌برازش ایجاد می‌کند.



شکل ۳. ساختار کلی کد پایتون و مدل Conv2D

### ۳.۳. ساختار مدل مبتنی بر Transformer

مدل Transformer بر پایه مکانیزم توجه (Attention) عمل کرده و قادر است وابستگی‌های زمانی بلندمدت موجود در سیگنال‌های جریان موتور را به صورت مستقیم مدل‌سازی کند. شکل ۴ بصورت فلوچارت گویای همه مراحل اجرای کد پایتون و ساختار کلی برنامه می‌باشد. مراحل پیش پردازش مدل ترنسفورمر مشابه مدل Conv2D است.



شکل ۴. نمایش ساختار کلی کد پایتون و مدل Transformer بصورت فلوچارت

### ۴.۳. مدل مبتنی بر CNN+LSTM

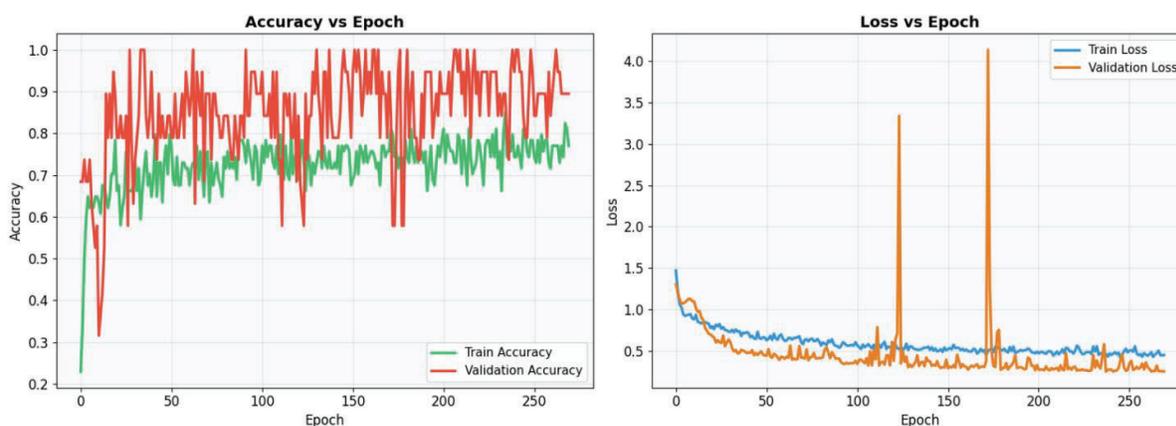
در این پیاده‌سازی، از یک مدل دوشاخه استفاده شده است. داده‌های ورودی از دو مسیر مجزا پردازش می‌شوند تا هم اطلاعات خام سیگنال و هم ویژگی‌های مهندسی‌شده به‌صورت هم‌زمان در فرآیند تصمیم‌گیری مدل لحاظ شوند. در شاخه اول، سیگنال خام به‌عنوان ورودی دریافت شده و با عبور از یک ساختار ترکیبی CNN+LSTM، ویژگی‌های زمانی و وابستگی‌های کوتاه‌مدت و بلندمدت سیگنال به‌صورت خودکار استخراج می‌شوند. این شاخه مسئول یادگیری الگوهای پنهان و غیرخطی موجود در داده‌های خام است. در شاخه دوم، ویژگی‌های مهندسی‌شده که به‌صورت دستی از سیگنال استخراج شده‌اند، مورد استفاده قرار می‌گیرند. این ویژگی‌ها شامل شاخص‌هایی نظیر Zero-Crossing و پارامترهای هیورت (Hjorth) هستند که اطلاعات آماری و دینامیکی مهمی از سیگنال ارائه می‌دهند. این ویژگی‌ها پس از استخراج، به یک شبکه تمام‌متصل (Fully Connected) اعمال شده و بازنمایی مناسبی از آن‌ها تولید می‌شود. در ادامه، خروجی دو شاخه با یکدیگر تلفیق (Fusion) شده و به بخش نهایی طبقه‌بندی‌کننده ارسال می‌شود. انگیزه اصلی استفاده از این معماری ترکیبی آن است که شباهت میان برخی کلاس‌ها، به‌ویژه کلاس‌های ۰ و ۱، بسیار زیاد بوده و تفاوت‌های میان آن‌ها بسیار ظریف است. در چنین شرایطی، استفاده صرف از مدل‌های یادگیری عمیق مبتنی بر سیگنال خام (مانند CNN+LSTM) به‌تنهایی برای تفکیک دقیق این کلاس‌ها کافی نیست. با وارد کردن ویژگی‌های مهندسی‌شده و وادار کردن مدل به استفاده هم‌زمان از این اطلاعات، دقت طبقه‌بندی به‌طور محسوسی افزایش یافته است. برای پیاده‌سازی این مدل از کتابخانه PyTorch استفاده شده است. در نتیجه، فرآیند آموزش شبکه به‌صورت دستی پیاده‌سازی شده و شامل اجرای صریح مراحل Forward Pass، محاسبه تابع هزینه، انجام Backward Pass و به‌روزرسانی پارامترها توسط بهینه‌ساز است. این رویکرد امکان ترکیب چند تابع هزینه مختلف و اعمال کنترل دقیق‌تر بر فرآیند آموزش را فراهم می‌کند؛ امکانی که در توابع آماده نظیر fit در کتابخانه‌هایی مانند TensorFlow و Keras یا در دسترس نیست یا پیاده‌سازی آن با پیچیدگی همراه است. آموزش و اجرای مدل بر روی پردازنده گرافیکی NVIDIA A100 انجام شده است که زمان اجرای کل فرآیند آموزش حدود ۲۵ دقیقه بوده است. با استفاده از این معماری دوشاخه و راهبرد ترکیبی یادگیری عمیق و مهندسی ویژگی، مدل نهایی

توانسته است به دقت طبقه‌بندی بالاتر از ۹۲ درصد دست یابد که نشان‌دهنده کارایی مناسب روش پیشنهادی در تفکیک کلاس‌های با شباهت بالا است.

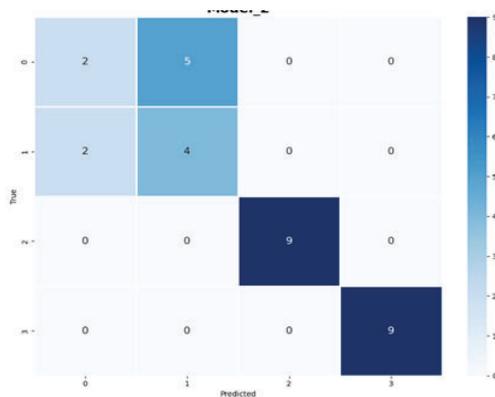
#### ۴. بررسی و مقایسه خروجی مدل‌ها

##### ۱.۴ مدل Conv1D

شکل ۵ نمودار دقت در آموزش و ارزیابی مدل Conv1D را مشاهده می‌کنیم. اگرچه عملکرد تابع هزینه در شکل ۵الف قابل قبول است و روند کاهشی خوبی دارد، ولی دقت آموزش مدل از تکرار ۵۰ به بعد، همانطور که در ۵الف مشاهده می‌شود دچار بیش‌برازش شده است. شکل ۵ب ماتریس در هم ریختگی نشان‌دهنده این موضوع است که مدل در تشخیص کلاس‌های ۰ و ۱ عملکرد قابل قبولی ندارد. ۷۷,۴۸ درصد دقت در دیتای تست نتیجه آموزش بر روی این مدل بوده است.



##### ۵. الف



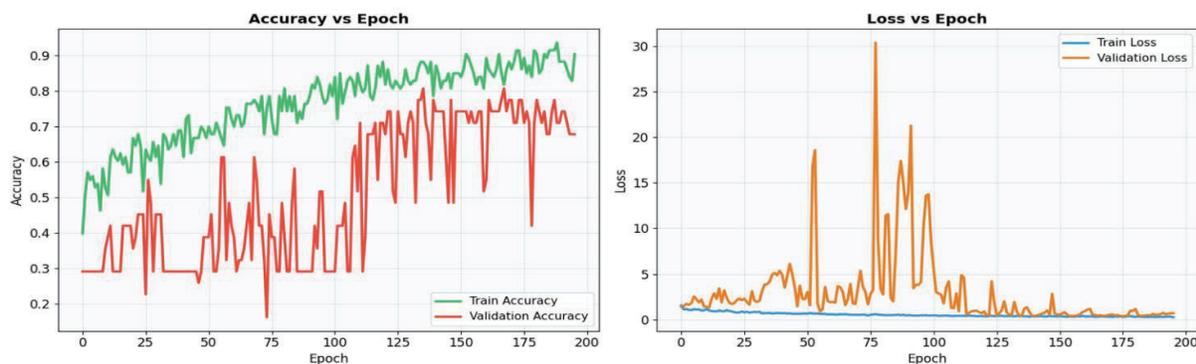
##### ۵. ب

شکل ۵. الف. نمودار تابع هزینه، ارزیابی و دقت مدل Conv1D. ب. ماتریس در هم ریختگی

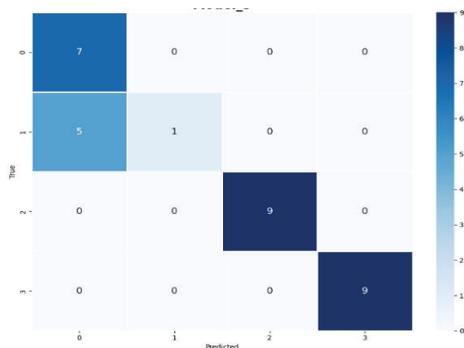
##### ۲.۴ مدل Conv2D

در شکل شماره ۶ نمودارهای آموزش و ارزیابی گویای این است که اگرچه در قسمت آموزش مدل عملکرد خوبی داشته است ولی عملکرد آن بر روی داده‌های ارزیابی پایداری خوبی ندارد. اما در نهایت از شکل جدول ماتریس در هم ریختگی می‌توان دریافت که در تشخیص

کلاس ها عملکرد بهتری داشته است. و در دقت نهایی در زمان تست درصد بالاتری بخود اختصاص داده است. ۸۳,۵ درصد، دقت در دیتای تست، برای این مدل بدست آمده است. که نسبت به مدل قبلی پیشرفت قابل ملاحظه ای است.



۶. الف

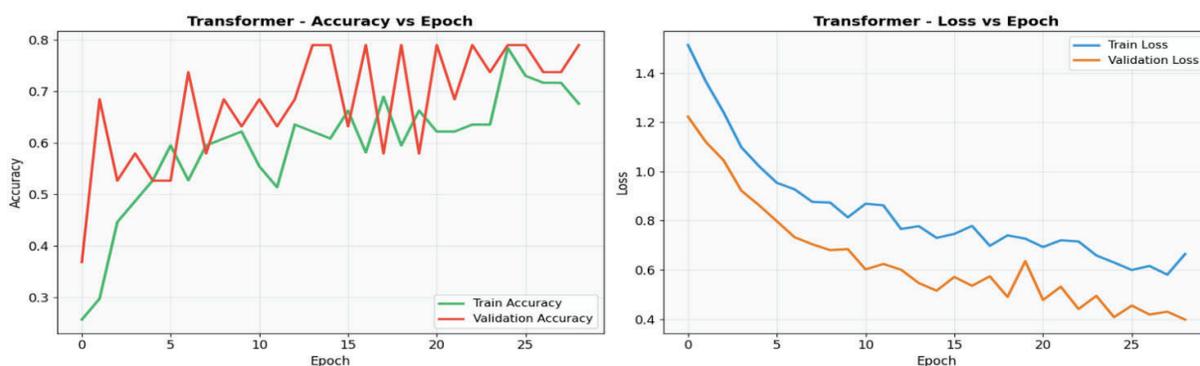


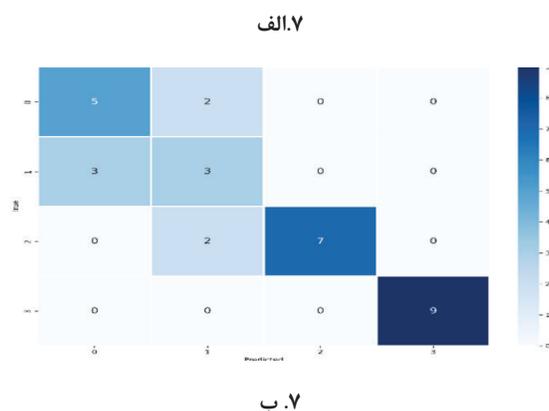
۶. ب

شکل ۶. الف. نمودار تابع هزینه، ارزیابی و دقت مدل Conv2D ب. ماتریس در هم ریختگی

### ۳,۴ مدل transformer

با استفاده از مدل ترنسفورمر به دقت ۸۰ درصد در دیتای تست رسیده ایم. همانطور که از شکل ۷ هم پیداست، مدل در یافتن پاسخ صحیح برای نمونه ها عملکردی بسیار قویتر دارد. ولی از طرف دیگر چهار بیش برآزش در دیتای تست و دقت کمتر از مدل هایی است که بررسی کردیم.





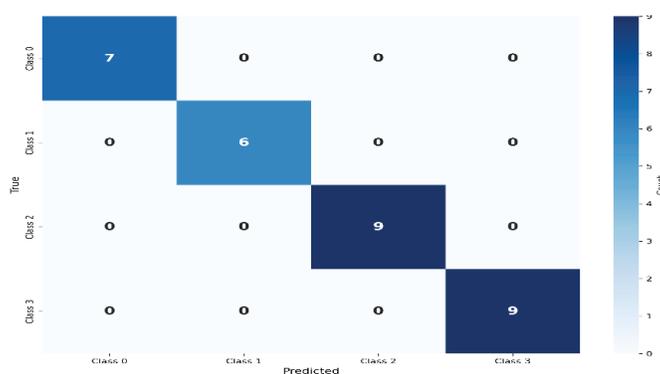
شکل ۷. الف. نمودار ارزیابی و دقت مدل Conv2D ب. نمودار تابع هزینه مدل ج. ماتریس در هم ریختگی

#### ۴.۴ مدل CNN+LSTM

بهبود نتایج در شکل ها و نمودارهای شکل ۸ کاملاً قابل مشاهده است. هر چه مطالعات بیشتری بر روی مدل و هدف مورد نظر انجام میگیرد به خروجی های بهتر و پایدارتری دست پیدا کرده ایم. میانگین دقت ۹۶.۵۷ درصد بر روی همه کلاس ها نتیجه بسیار قابل قبولی برای این پژوهش می باشد. شکل ۵ نمودار آموزش، تابع هزینه و ماتریس در هم ریختگی را نشان می دهد.



شکل ۸. الف



شکل ۸. ب

شکل ۸. نمودار تابع هزینه، آموزش و ارزیابی مدل CNN+LSTM ب. ماتریس در هم ریختگی

## ۵. نتیجه گیری

در این پژوهش، طبقه‌بندی چندحالتی خطاهای سنسور اثر هال در موتور BLDC با استفاده از سیگنال‌های جریان و مدل‌های مختلف یادگیری عمیق مورد بررسی قرار گرفت. نتایج نشان داد که اگرچه مدل‌هایی مانند CNN یک‌بعدی، CNN دوبعدی و Transformer قادر به استخراج الگوهای مفید از داده‌ها هستند، اما مدل ترکیبی **CNN+LSTM** با بهره‌گیری هم‌زمان از استخراج ویژگی‌های مکانی و مدل‌سازی وابستگی‌های زمانی، عملکرد برتری از خود نشان داده و به دقت ۹۳٪ در داده‌های آموزش و ۹۱٪ در داده‌های آزمون دست یافته است. همچنین مشخص شد که انتخاب معماری مناسب، در کنار پیش‌پردازش هدفمند و استفاده از مهندسی ویژگی، نقش کلیدی در بهبود کارایی سامانه‌های تشخیص خطا دارد. یافته‌های این پژوهش می‌تواند مبنای توسعه سیستم‌های پایش وضعیت هوشمند و قابل‌اعتماد برای موتورهای BLDC در کاربردهای صنعتی باشد.



## ۶. منابع

- [1] Kenny SauKangChu, KuewwaiChew and YoongChoonChang, "Fault-Diagnosis and Fault-Recovery System of Hall Sensors in Brushless DC Motor Based on Neural Networks".
- [2] Jain, C.; Garg, P.; Jain, A. Hall-Effect Sensor Fault Diagnosis, Identification and Remedial Strategy in Permanent Magnet Brushless DCDrive; Ieeexplore.ieee.org. 2021. Available online: <https://ieeexplore.ieee.org/document/8707593/>(accessed on 16 July 2021).
- [3] His Kao, Wei-Jen Wang, Yi-Horng Lai, and Jau-Woei Perng , "Analysis of Permanent Magnet Synchronous Motor Fault Diagnosis Based on Learning". Digital Object Identifier 10.1109/TIM.2018.2847800
- [4] FRANÇOIS CHOLLET, "Deep Learning with Python" . SECOND EDITION
- [5] Seul Ki Hong, and Yongkeun Lee Optimizing Detection, "Compact MobileNet Models for Precise Hall Sensor Fault Identification in BLDC Motor Drives". date of publication 31 May 2024, Department of Semiconductor Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
- [6] YONGKEUNLEE AND STEPHENMAKONIN, " BLDC Hall Sensor Displacement Dataset (BLDC-HSD)". Date of publication 8 October 2024. DATA DOI/PID 10.21227/17e3-t177
- [7] S. Ki Hong and Y. Lee, "Optimizing Detection: Compact MobileNet Models for Precise Hall Sensor Fault Identification in BLDC Motor Drives," in *IEEE Access*, vol. 12, pp. 77475-77485, 2024, doi: 10.1109/ACCESS.2024.3407766.
- [8] K.S., K. V., & J, G. (2023). Hall sensor fault diagnosis and compensation in Brushless DC motor drive using artificial intelligence technique. *International Journal of Circuit Theory and Applications*, 51, 2777–2797. <https://doi.org/10.1002/cta.3538>
- [9] Natarajan, S., & Jothimani, G. (2023). Hall sensor fault diagnosis and compensation in Brushless DC motor drive using artificial intelligence technique. *International Journal of Circuit Theory and Applications*, 51, 2777–2797. <https://doi.org/10.1002/cta.3538>
- [10] Gallon, R., Schiemenz, F., Menicucci, A., & Gill, E. (2024). Convolutional Neural Network Design and Evaluation for Real-Time Multivariate Time Series Fault Detection in Spacecraft Attitude Sensors. <https://doi.org/10.48550/arxiv.2410.09126>
- [11] T. Marković, A. Dehlaghi-Ghadim, M. Leon, A. Balador, and S. Punnekkat, "Time-series Anomaly Detection and Classification with Long Short-Term Memory Network on Industrial Manufacturing Systems," pp. 171–181, Sept. 2023, doi: 10.15439/2023f5263.
- [12] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data," *Proceedings of The Vldb Endowment*, vol. abs/2201.07284, Jan. 2022.
- [13] YONGKEUN LEE, "BLDC HALL SENSOR DISPLACEMENT", *IEEE Dataport*, September 28, 2024, doi:10.21227/17e3-t177