



به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر



درس شبکه‌های عصبی و یادگیری عمیق

تمرین چهارم

نام دانشجو	ثمین سلوکی	
شماره دانشجویی	۸۱۰۹۰۳۰۵۰	

فهرست

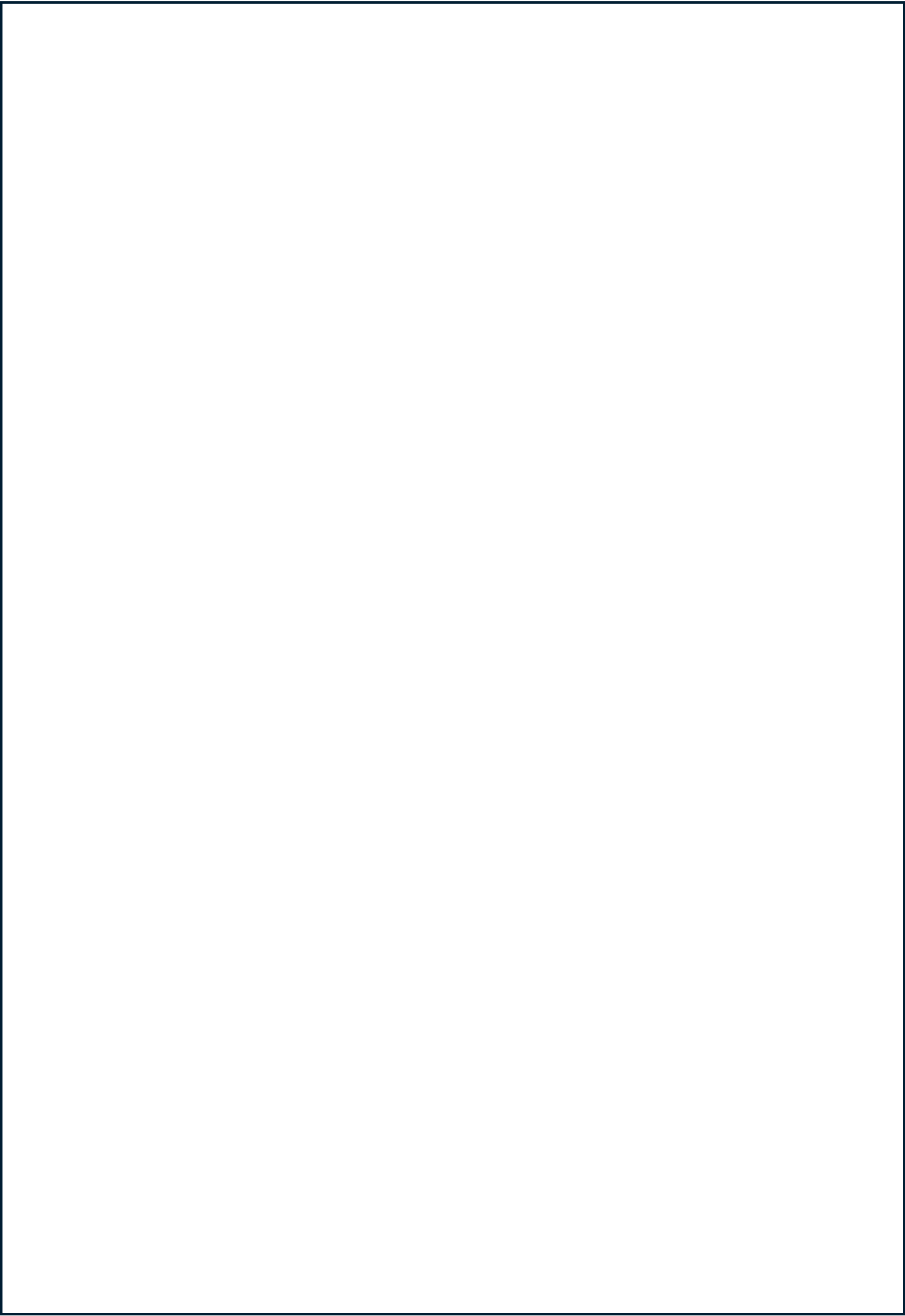
- پرسش (سامانه های سایبرفیزیکی: نگهداری هوشمند ۲
- ۲-۱) پیش پردازش داده ها ۲
- ۲-۲) مدلسازی و ارزیابی ۶
- ۲-۳) مقایسه با مدل های پایه ۱۱

شکل‌ها

7 Life time of each engine -7 Figure
7 RUL محاسبه ی -8 Figure
۸ Data Selection نمودار تغییر مقادیر برای انجام فاز -9 Figure
۹ Timing windows+Data Labeling-10 Figure
۱۰ مدل پیشنهادی -طبقه بندی -11 Figure
۱۱ CNN+LSTM: classification-with early stopping -12 Figure
۱۱ -CNN+LSTM: classification-without early stopping-۱۳ Figure
۱۱ - confusion matrix : - CNN+LSTM: classification with early stopping -۱۴ Figure
۱۱-confusion matrix:CNN+LSTM: classification without early stopping -۱۵ Figure
۱۱ - ROC : - CNN+LSTM: classification-with early stopping ۱۶ Figure
۱۱ - ROC:CNN+LSTM: classification-without early stopping ۱۷ Figure
	CNN+LSTM:Reg-with CNN+LSTM: Regression-without early stopping -۱۸ Figure
۱۵	early stopping
۱۵ CNN+LSTM: Regression-without early stopping-۱۹ Figure
۱۹ CNN پایه مدل بوسیله RUL بنی شده و پیش بینی و مقدار مقادیر واقعی -2۰ Figure
۱۴ CNN پایه مدل بوسیله طبقه بندی و هم‌ریختگی -2۱ Figure
۱۴ CNN پایه مدل بوسیله طبقه بندی و هم‌ریختگی -2۲ Figure
۲۱ LSTM پایه مدل بوسیله RUL بنی شده و پیش بینی و مقدار مقادیر واقعی -2۳ Figure
۲۲ LSTM پایه مدل بوسیله طبقه بندی و هم‌ریختگی -2۴ Figure

جدول‌ها

۵ CNN-LSTM, CNN, LSTM برای سه مدل معیارهای ارزیابی -جدول ۱
---	--



پرسش (سامانه های سایبرفیزیکی: نگهداری هوشمند

گسترش سیستم های فیزیکی سایبری (CPS) و پیشرفت فناوری های اینترنت اشیا (IoT) چشم انداز امیدوارکننده ای را برای قابلیت اطمینان بالا، در دسترس بودن، قابلیت نگهداری و فرآیند تولید ایمنی ارائه می کند. در موضوع نگهداری و تعمیرات، نگهداری و تعمیرات پیشگیرانه (PdM) قابلیت و پتانسیل بالایی در صنعت ۴ از خود نشان داده است. PdM را می توان به عنوان «روشی هوشمند برای به حداکثر رساندن در دسترس بودن یک ماشین» توصیف کرد و به استراتژی های تشخیص و پیش بینی زودهنگام شرایط و انجام اقدامات فنی لازم برای تشخیص و پیش بینی این موارد و نتیجه کاهش هزینه های نگهداری، بهبود عملکرد عملیاتی و حمایت از تصمیم گیرنده اشاره دارد. در این حوزه، پیش بینی عمر مفید باقی مانده (RUL) هسته PdM است. نتایج ارزیابی دقیق و قابل اعتماد RUL اطلاعات ارزشمندی را در اختیار تصمیم گیرندگان قرار می دهد تا استراتژی تعمیر و نگهداری مناسب را برای به حداکثر رساندن استفاده از تجهیزات و جلوگیری از خرابی پرهزینه و در نتیجه کاهش قابل توجه هزینه های تعمیر و نگهداری در اختیار بگیرند.

هدف این پروژه اجرای یک معماری چند وظیفه ای یادگیری عمیق ترکیبی کاربردی با ادغام مزایای CNN و LSTM برای انعکاس ارتباط پیش بینی عمر مفید باقیمانده با فرآیند تشخیص وضعیت سلامت است. در این معماری پیشنهادی، CNN به عنوان استخراج کننده ویژگی برای فشرده سازی داده های سری زمانی پایش وضعیت و استخراج مستقیم ویژگی های مکانی-زمانی مهم از داده های ورودی چندحسی خام استفاده می شود و LSTM برای ثبت ویژگی های وابستگی موقت درازمدت استفاده می شود.

۱-۲) پیش پردازش داده ها

داده های این پروژه مربوط به داده های تخریب موتور توربوفن از شبیه سازی سیستم پیشران هوا (C-MAPSS) است که توسط مرکز تعالی پیش بینی در مرکز تحقیقاتی ناسا جمع آوری شده. این دیتاست شامل خوانش حسگر از تعدادی از موتورهای توربین گازی شبیه سازی شده هواپیما است که به صورت سری زمانی چند متغیره ثبت شده است. همه موتورهای مذکور از یک نوع هستند، اما هر موتور درجات مختلف سایش اولیه و تغییرات در طول تولید دارد (که این مقادیر ناشناخته است و در کل موتور سالم در نظر گرفته می شود). دیتاست شامل مجموعه داده های زیر است:

- مجموعه آموزش: داده های چرخش تا خرابی موتور هواپیما.
- مجموعه تست: داده های عملیاتی موتور هواپیما بدون ثبت رخداد های خرابی.
- داده های Ground truth: اطلاعات چرخه های باقی مانده واقعی برای هر موتور در داده تست.

دیتای آموزش و دیتای تست دارای ۲۶ ستون است: شماره موتور، گام زمانی، سه تنظیمات عملیاتی و ۲۱ اندازه گیری سنسور.

داده های ورودی مدل، توالی ای از داده های سنسورهای موتور ها در سیکل های متوالی تا شکست موتور است که بر اساس پنجره زمانی منتخب ایجاد می شوند. در ادامه درباره ی پنجره زمانی بیشتر توضیح خواهیم داد.

برای پیش پردازش انجام شده ابتدا به توضیح مفاهیم و بعد به توضیح مراحل اجرایی در کد می پردازیم: در پیش پردازش داده ها ابتدا اقدام به Data selection شد: با بررسی نمودارهای خوانش سنسور در طول زمان، مشخص است که برخی از این ۲۴ مورد، در طول زمان تغییرات قابل توجهی ندارند، بدین دلیل حضورشان واریانس خاصی را توضیح نخواهد داد، به عبارتی حضور این دیتاها اطلاعات ارزشمندی در مورد طول عمر باقی مانده به ما نمیدهد و صرفا باعث افزایش پیچیدگی شبکه ی عصبی می شوند. این ویژگی ها از دیتاست حذف شدند:

```
'operational setting 3','sensor measurement 1',  
'sensor measurement 5' , 'sensor measurement 10'  
'sensor measurement16','sensor measurement18','sensor measurement19',
```

قدم بعدی، Data Normalization است: داده های این دیتاست از چندین منبع متفاوت تغذیه شده اند که دارای دامنه ی متفاوت مقادیر می باشند، با هدف هم مقیاس کردن داده ها اقدام به min-max normalization می کنیم تا تمامی اعداد در محدوده ی ۰ تا ۱ قرار گیرند.

قدم بعدی، Data labeling است: برای بخش رگرسیون، آخرین داده ی ورودی (زمان مربوطه) زمانی است که موتور از کار ایستاده یا به عنوان [ناسالم] اعلام شده. در این صورت عمر عملیاتی موتور برابر است با $T^{(i)}$ که $T^{(i)}$ برابر است با طول سری زمانی دیتای سنسور موتور مربوطه (=زمان آخرین داده ی ورودی)

برای مسئله ی طبقه بندی، داده ها به دو کلاس تقسیم شدند، بدین صورت که کلاس ۰ شامل موتور هایی است که عمر باقی مانده ی شان بیشتر از پنجره زمانی است، این دسته شامل دستگاه های سالم است. کلاس ۱ شامل موتور هایی است که عمر باقی مانده ی شان از پنجره زمانی بیشتر نباشد، این دسته شامل دستگاه های ناسالم است.

قدم بعدی، انجام Time-window Processing است: در این مرحله با هدف ایجاد ورودی sequential، پنجره ای بر داده های خام حرکت می دهیم. طول این پنجره ی زمانی، N_L است. sliding stride برای این پنجره طبق گفته ی مقاله ی پایه ۱ انتخاب شد. نهایتا ابعاد ورودی داده ها

به شکل $N_L * N_F$ در می آید که برابر N_L طول پنجره ی زمانی و N_F برابر با تعداد ویژگی هاست. برای شفافیت بیشتر به این نکته اشاره می کنیم که نمونه های یک پنجره ی زمانی مربوط به یک موتور است و کلاس تارگت آخرین داده به عنوان برچسب تارگت پنجره ی زمانی مربوطه اعلام می شود.

در کد و مراحل اجرایی، پس از فراخوانی داده ها در دیتا فریم، و بررسی ابعاد و نوع داده و مقادیر و گذر از بخش فهم داده به سراغ آماده سازی داده رفتیم. در ابتدا سعی کردیم برای هر موتور (براساس مقادیر ستون `Unitnumber`) طول عمر موتور را بدستبیاوریم و ستونی تشکیل دهیم که نشان دهنده ی عمر موتور باشد و این چنین ستون `life_time` را تشکیل دادیم. این ستون در ادامه برای محاسبه ی RUL هر پنجره استفاده شد. همچنین نمونه ای از گزارش طول عمر برخی از موتورها در ادامه آمده است:

```
life time of engine Num.1= 192
life time of engine Num.2= 287.0
life time of engine Num.3= 179.0
life time of engine Num.4= 189.0
life time of engine Num.5= 269.0
life time of engine Num.6= 188.0
life time of engine Num.7= 259.0
life time of engine Num.8= 150.0
life time of engine Num.9= 201.0
life time of engine Num.10= 222.0
life time of engine Num.11= 240.0
life time of engine Num.12= 170.0
life time of engine Num.13= 163.0
life time of engine Num.14= 180.0
life time of engine Num.15= 207.0
life time of engine Num.16= 209.0
life time of engine Num.17= 276.0
life time of engine Num.18= 195.0
life time of engine Num.19= 158.0
life time of engine Num.20= 234.0
```

Life time of each engine -۷ Figure

سپس ستون RUL محاسبه شد به شکل زیر:

```
train_data['RUL']= train_data['life_time']-train_data['time in cycles']
train_data['RUL'] = [130 if x > 130 else x for x in train_data['RUL']]
```

RUL -۸ محاسبه ی Figure

در سطر اول کد مشاهده می کنید که RUL هر موتور از حاصل تفریق ستون طول عمر و سایکل بدست آمده، و در ادامه طبق گفته ی مقاله، مقادیر بیشتر از ۱۳۰ به ۱۳۰ محدود شده اند.

در مرحله ی بعد، انتخاب داده انجام شد. نمودار تغییر مقادیر در طی زمان کشیده شد و طبق گفته ی مقاله، متغیر هایی که نزدیک به ثابت بودند حذف شدند. نمودار تغییرات مربوطه را در زیر مشاهده می کنید:

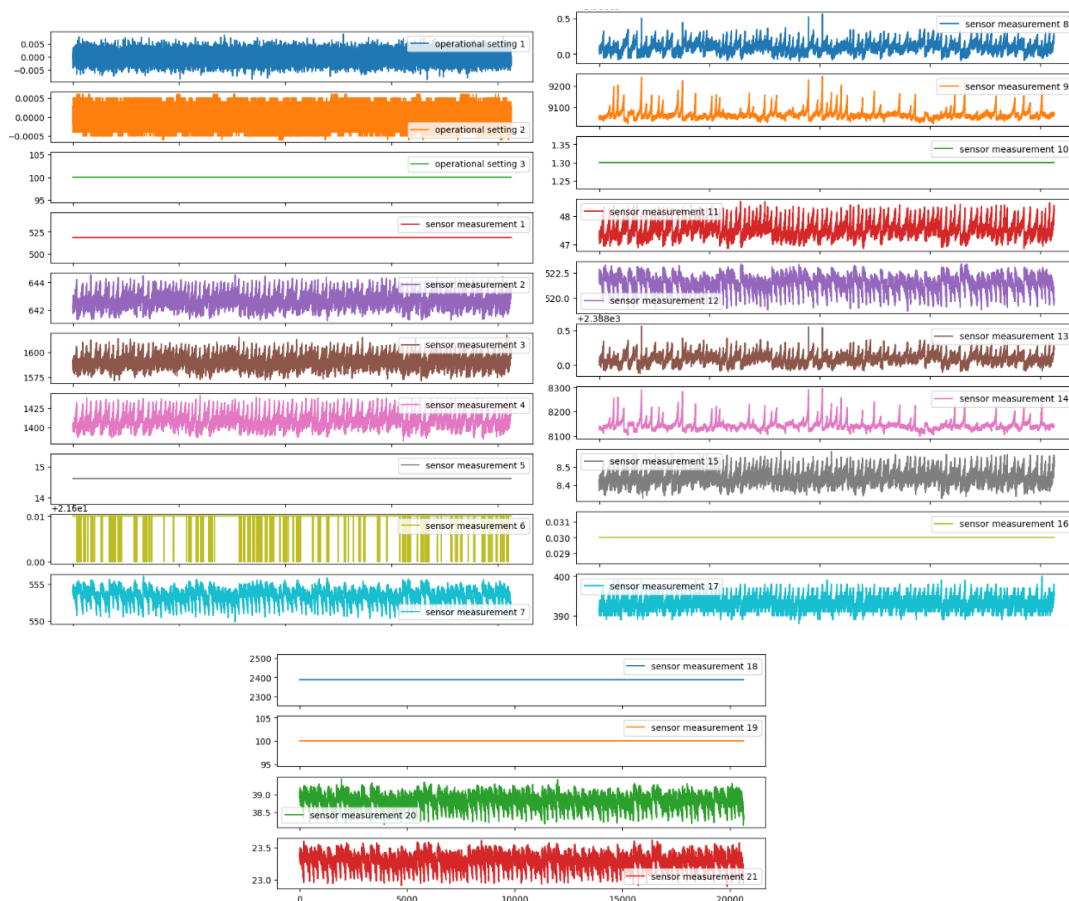


Figure 9: نمودار تغییر مقادیر برای انجام فاز Data Selection

بنابراین ستون های زیر از دیتاست حذف شدند:

```
'operational setting 3','sensor measurement 1',
'sensor measurement 5', 'sensor measurement 10'
'sensor measurement16','sensor measurement18','sensor measurement19',
```

برای انجام فاز نرمالسازی از MinMaxScaler از کتابخانه ی sklearn استفاده کردیم.

مرحله ی بعد، برچسب گذاری را انجام دادیم. این مرحله در کد ما با مرحله ی پنجره ی زمانی ادغام شده است. توضیحات مربوطه بدین صورت است: برای محاسبه ی پنجره ی زمانی، طبق گفته ی مقاله طول پنجره ۳۰ با stride=1 انتخاب شد. این انتخاب پنجره بدین صورت انجام شد که ۳۰ ردیف - ۳۰ ردیف داده ها با stride=1 انتخاب شد که هر کدام از این پنجره ها به طول ۳۰ یک sequence را تشکیل میدهند، این stride را به صورت حرکت در راستای عمود دیتاست در نظر بگیرید.

در ادامه برچسب گذاری برای رگرسیون و طبقه بندی انجام شد بدین صورت که:

برای رگرسیون: RUL آخرین دیتا ی پنجره ی منتخب به عنوان RUL برای sequence اعلام شد.

برای طبقه بندی: اگر RUL برای آخرین دیتای sequence بیشتر از ۳۰ (طول پنجره-افق) بود یعنی موتور در آن بازه خراب نشده و برچسب ۰ می‌گرفت و در غیر این صورت ۱ که برچسب خرابی بود. کد مربوطه به شکل زیر است:

```
X=[]
y_reg=[] # label for Regression to predict RUL
y_cl=[] # label for Classification to predict healthy(0) vs. failure(1)
SEQ_LEN =30 #Sequence length(NL)
for i in range(SEQ_LEN,train2_normalized.shape[0]):
    X.append(train2_normalized.iloc[i-SEQ_LEN:i,:-1])
    y_reg.append(train2_normalized.at[i,'RUL'])
    tar= 0 if train2.at[i,'RUL'] > SEQ_LEN else 1
    y_cl.append(tar)
X=np.array(X)
y_reg=np.array(y_reg)
y_cl=np.array(y_cl)
```

Figure ۱۰-۱ Timing windows+Data Labeling

لازم به ذکر است که تمامی مراحل فوق برای داده های آموزش و تست انجام شده است. برای محاسبه RUL دیتای تست، بیشترین مقدار سایکل موتور را با مقدار متناظر از فایل RUL_FD001 جمع کردیم و از مقدار ستون سایکل سطر منها کردیم.

۲-۲) مدلسازی و ارزیابی

در این بخش بر اساس مدل پیشنهادی مقاله، ترکیب هایبرید LSTM+CNN برای پیش بینی RUL و طبقه بندی باینری برای پیش بینی شرایط سلامت موتور اجرا شد. مدل پیشنهادی از ۳ بخش اصلی شبکه های عصبی کانولوشنی، شبکه LSTM و لایه های fully connected است. در این معماری، CNN برای جلوگیری از مشکل معرفی شده است، که می تواند بعد داده های اصلی را کاهش دهد و ویژگی های مهم را استخراج کند. در ابتدا، داده های اجرا تا شکست ضبط شده و به چند دنباله تقسیم می شود. فیلتر CNN این معماری تک بعدی است. در ادامه ی مدل از LSTM برای پردازش وابستگی های زمانی داده ها استفاده شده است. در مدل های توسعه یافته در مقاله برای جلوگیری از بیش پردازش داده ها، از Dropout و تکنیک Early stopping استفاده شد. لازم به ذکر است که تکنیک Early stopping بر ۱۰ درصد داده ها و با معیار توقف عدم بهبود loss انجام شده. در مدل ارایه شده طبق مقاله ی پایه تابع هزینه، تابع Mean Square Error است و بهینه ساز ادام و RMSProp انتخاب شده. مدل ارایه شده ی مقاله را برای رگرسیون و تخمین RUL و برای طبقه بندی و پیش بینی وضعیت سلامت

موتور اجرا کردیم. مدل های پیشنهادی را یک بار با Early stopping و یک بار بی Early stopping هم برای مدل رگرسیون و هم برای مدل طبقه بندی اجرا کردیم. معماری مدل به شرح زیر است:

```
model_cl = Sequential()

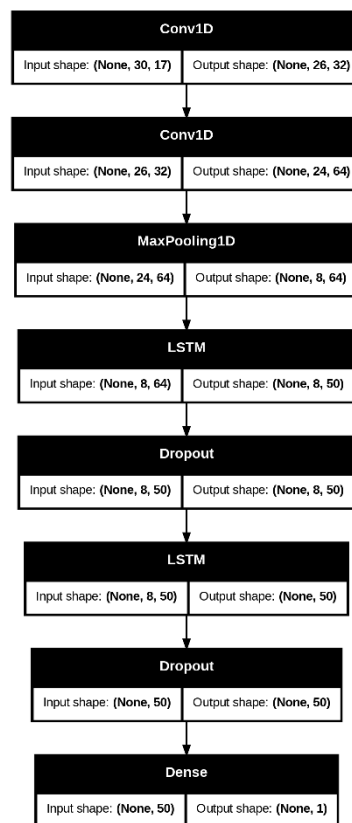
model_cl.add(Conv1D(filters=32, kernel_size=5, activation='relu', input_shape=X.shape[1:]))
model_cl.add(Conv1D(filters=64, kernel_size=3, activation='relu'))

model_cl.add(MaxPooling1D(pool_size=3))

model_cl.add(LSTM(units=50, return_sequences=True))
model_cl.add(Dropout(rate=0.2))
model_cl.add(LSTM(units=50))
model_cl.add(Dropout(rate=0.2))
model_cl.add(Dense(units=1, activation='sigmoid'))

model_cl.compile(optimizer=Adam(), loss='binary_crossentropy')
```

Figure ۱۱- طبقه بندی -مدل پیشنهادی



به عبارت دیگر معماری مدل:

مدل را با جزییات گفته شده در مقاله آموزش دادیم برای فیت کردن مدل هم اعتبار سنجی ده درصد داده ها- در طی ۱۰۰ اپاک انجام شد.

نتایج عملکرد این مدل بر داده ی تست را در زیر مشاهده میکنید. ستون سمت راست، نتایج آموزش دیده شده با توقف زودهنگام در ۱۲ اپاک است و ستون سمت چپ، نتایج آموزش در ۱۰۰ اپاک می باشد.

	precision	recall	f1-score	support
0	0.99	0.92	0.95	12734
1	0.19	0.75	0.31	332
accuracy			0.91	13066
macro avg	0.59	0.83	0.63	13066
weighted avg	0.97	0.91	0.94	13066

Figure ۱۳ -CNN+LSTM: classification-without early stopping

	precision	recall	f1-score	support
0	0.99	0.90	0.95	12734
1	0.16	0.69	0.26	332
accuracy			0.90	13066
macro avg	0.57	0.80	0.60	13066
weighted avg	0.97	0.90	0.93	13066

Figure ۱۲ -CNN+LSTM: classification-with early stopping

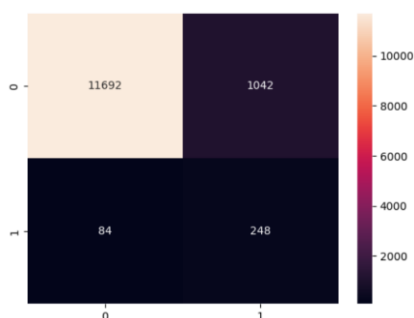


Figure ۱۵- confusion matrix: CNN+LSTM: classification- without early stopping

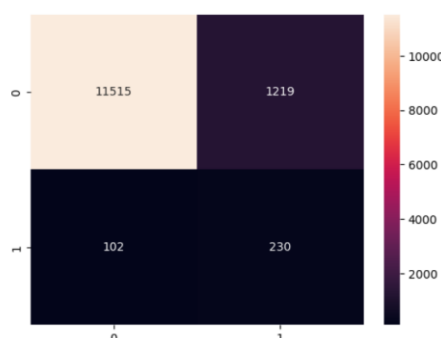


Figure ۱۴- confusion matrix : - CNN+LSTM: classification- with early stopping

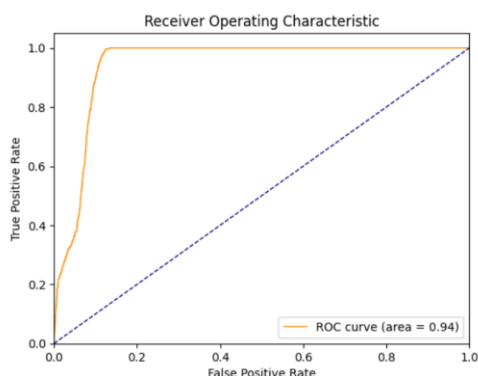


Figure ۱ - ROC: CNN+LSTM: classification-without early stopping

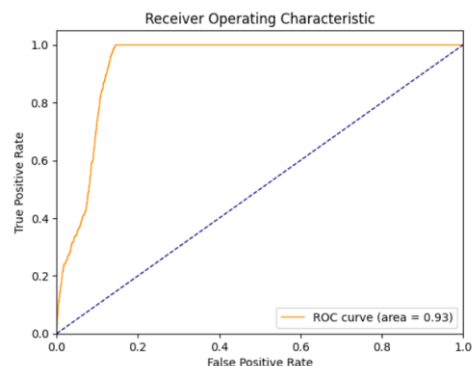


Figure ۱۶- ROC : - CNN+LSTM: classification-with early stopping

بر اساس ماتریس های سردرگمی مشخص است که عملکرد هر دو مدل (با و بدون توقف اولیه) در طبقه خوب بوده است. مدل بدون توقف زودهنگام در مقایسه با مدل توقف زودهنگام (۸۴) به مقدار واقعی مثبت (TP) 11692 دست یافته؛ این نشان می دهد که این مدل توانسته است تعداد بیشتری از موتورهای سالم (کلاس ۰) را به درستی طبقه بندی کند. از طرف دیگر، مقدار مثبت کاذب (FP) بالاتر ۱۲۱۹ در مقایسه با ۲۴۸ با توقف زودهنگام دارد. این به این معناست که این مدل موتورهای معیوب بیشتری (کلاس ۱) را

به اشتباه به عنوان موتورهای سالم طبقه بندی کرد. همچنین مقادیر منفی کاذب (FN) ۱۰۴۲ با این مدل در مقایسه با ۲۳۰ با توقف زود هنگام بیشتر بوده است. این نشان می دهد که مدل، دسته بندی موتورهای معیوب بیشتر (کلاس ۱) به عنوان موتورهای سالم (کلاس ۰) را از دست داده است.

به طور کلی به نظر می رسد مدل با توقف زود هنگام به تعادل بهتری بین مثبت های واقعی و مثبت های کاذب دست یافته است در حالی که مدل بدون توقف زود هنگام به نرخ مثبت واقعی بالاتر اما به قیمت مثبت های کاذب بیشتر دست یافت. به نظر می رسد توقف زود هنگام با جلوگیری از برازش بیش از حد بر روی داده های آموزشی، به تعمیم بهتر مدل در داده های دیده نشده کمک کرده است.

با توجه به دیگر متریک ها مدل بدون توقف زود هنگام دارای Precision (کلاس ۰): ۰.۹۹ است که نشان می دهد از تمام موتورهای طبقه بندی شده به عنوان سالم (کلاس ۰) توسط این مدل، ۹۹٪ واقعا سالم بوده اند. Recall (کلاس ۰) = ۰.۹۲ به این معنی است که از تمام موتورهای سالم واقعی (کلاس ۰) در داده ها، مدل توانسته ۹۲٪ را به درستی طبقه بندی کند. امتیاز F1 (کلاس ۰) = ۰.۹۵ میانگین هارمونیک Precision و Recall است و هر دو معیار را در نظر می گیرد. امتیاز ۰.۹۵ تعادل خوبی بین دقت و یادآوری موتورهای سالم (کلاس ۰) را نشان می دهد. Precision (کلاس ۱) = ۰.۱۹ نشان می دهد که از تمام موتورهای طبقه بندی شده به عنوان معیوب (کلاس ۱) توسط این مدل، تنها ۱۹٪ واقعا معیوب بوده اند. این نشان از دقت پایین است، به این معنی که این مدل در دسته بندی موتورهای سالم به عنوان موتورهای معیوب اشتباهات زیادی داشته است Recall (کلاس ۱) = ۰.۷۵ به این معنی است که از تمام موتورهای معیوب واقعی (کلاس ۱) در داده ها، مدل توانسته ۷۵٪ را به درستی طبقه بندی کند. امتیاز F1 (کلاس ۱) = ۰.۳۱ امتیاز پایینی است به دلیل دقت پایین برای کلاس ۱ نشان می دهد که مدل در شناسایی صحیح موتورهای معیوب نیاز به بهبود دارد. نهایتا به نظر می رسد توقف زود هنگام تعمیم مدل را برای

موتورهای سالم (کلاس ۰) بهبود می بخشد، اما ممکن است تأثیر قابل توجهی بر طبقه بندی موتور معیوب (کلاس ۱) نداشته باشد.

نتایج مدل رگرسیون به شکل زیر است:

MSE: 0.12745198160522445
RMSE: 0.3570041758932582
MAE: 0.28646467237396867
MAPE: 0.3059437222906943%

CNN+LSTM: Regression-without early stopping

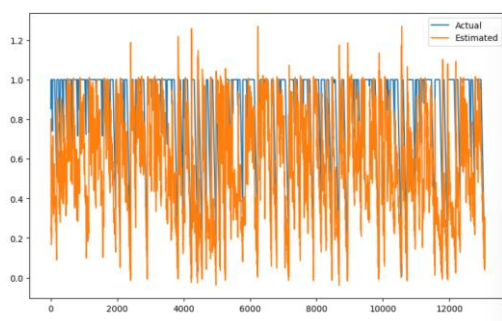


Figure-۱۹ CNN+LSTM: Regression-without early stopping

MSE: 0.11878733203785996
RMSE: 0.34465538155940634
MAE: 0.27589188000041354

CNN+LSTM: classification-with early stopping

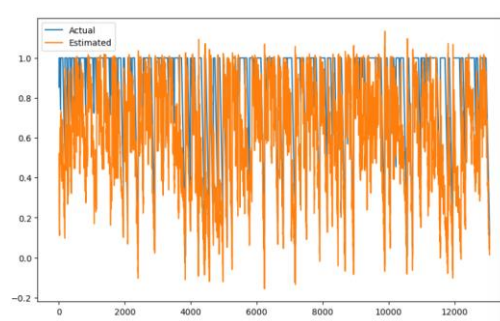


Figure -۱۸ CNN+LSTM: Regression-with early stopping

ستون سمت چپ نتایج مدل آموزش دیده بدون توقف زودهنگام است و نتایج سمت راست مربوط به مدل آموزش دیده با توقف زودهنگام در طی ۱۲ اپیاک است. همانطور که مشخص است $MSE=0.1274$ برای مدل بدون توقف و 0.1188 برای مدل با توقف اولیه. MSE پایین تر است و نشان دهنده عملکرد بهتر است، بنابراین در این مورد، مدل با توقف زودهنگام به MSE کمی پایین تر دست یافته. $RMSE=0.3570$ برای بدون توقف اولیه و 0.3446 برای با توقف زودهنگام. مشابه MSE ، $RMSE$ پایین تر نشان دهنده عملکرد بهتر است، و دوباره مدل توقف اولیه کمی بهتر کرده است. $MAE=0.2865$ برای بدون توقف زودهنگام و 0.2759 برای با توقف زودهنگام. MAE نشان دهنده اختلاف میانگین بین مقادیر RUL پیش بینی شده و واقعی است. MAE پایین تر نشان دهنده عملکرد بهتر است، بنابراین دوباره مدل توقف اولیه بهبود جزئی را نشان می دهد.

با بررسی نمودار تخمین زده شده بی توقف زود هنگام در برابر مقادیر واقعی دریافت می شود که مدل بی توقف زود هنگام گستره وسیع تری از نقاط داده را در اطراف خط مورب داشته که نشان می دهد خطاهای مدل پراکنده تر هستند. همچنین ممکن است برخی موارد پرت وجود داشته باشد که در آن پیش بینی مدل

به طور قابل توجهی از RUL واقعی منحرف شود. نمودار برای مدل با توقف زودهنگام به نظر می رسد طبقه بندی دقیق تری از نقاط داده در اطراف خط مورب دارد، که نشان می دهد خطاهای مدل کوچکتر هستند. بر اساس نمودار ها و معیارها، به نظر می رسد مدل CNN-LSTM با توقف زودهنگام برای کار تخمین RUL کمی قوی تر عمل می کند. مدل توقف اولیه به مقادیر MSE، RMSE و MAE کمتری دست یافت، که نشان می دهد به طور متوسط، پیش بینی های مدل به مقادیر واقعی RUL نزدیک تر بودند. نمودار پراکندگی برای مدل توقف اولیه نیز به صورت بصری یک طبقه بندی فشرده تر از نقاط داده در اطراف خط مورب را تأیید می کند.

۳-۲) مقایسه با مدل های پایه

در این مرحله با توجه به اینکه در مقاله مذکور، جهت بررسی نتایج مدل ارائه شده با مقالات مرتبط قبل دو مدل CNN و LSTM را ذکر کرده است با مراجعه به مقالات مذکور، هر دوی این معماری های پیاده سازی شده و در دیتاست تشکیل شده در بخش قبل اجرا شده اند و نتایج و متریک هایی که در بخش قبل در جهت ارزیابی عملکرد مدل هیبرید ارائه شده، در نظر گرفته شدند، برای این دو مدل نیز پیاده کردیم تا مقایسه ای بین عملکرد این سه مدل صورت پذیرد.

مدل CNN:

مدل ساخته شده در ابتدا شامل یک لایه کانولوشنی یک بعدی با تعداد ۸ فیلتر و سایز ۶ با تابع فعال ساز RELU و ابعاد ورودی ۱۷*۳۰ که همان سایز پنجره یا هرکدام از لیست های ورودی به مدل می باشد. در ادامه یک لایه maxpooling با سایز ۴ قرار می دهیم. در ادامه یک لایه کانولوشنی یک بعدی دیگر با ۱۴ فیلتر و سایز ۳ و تابع فعال ساز RELU است. در ادامه یک لایه maxpooling با سایز ۴ قرار می دهیم. سپس با دستور Flatten ویژگی های استخراج شده را به ترتیب می کنیم. و در ادامه سه لایه پرسپترون به ترتیب با تعداد نرون ۵۰، ۱۰ و ۱ به همراه تابع فعال سازی های relu برای دو لایه اول و تابع خطی برای لایه آخر در مدل رگرسیون و تابع sigmoid برای لایه آخر در مدل طبقه بندی قرار داده شده است. Optimizer از نوع RMSprop و تابع loss برای مدل رگرسیون از نوع mse و برای مدل طبقه بندی binary_crossentropy تعریف می شود. ساختار لایه های مدل مطابق شکل زیر است:

Model: "sequential"		
Layer (type)	Output Shape	Param #
=====		
conv1d (Conv1D)	(None, 25, 8)	824
max_pooling1d (MaxPooling1D)	(None, 6, 8)	0
conv1d_1 (Conv1D)	(None, 4, 14)	350
max_pooling1d_1 (MaxPooling1D)	(None, 2, 14)	0
flatten (Flatten)	(None, 28)	0
dense (Dense)	(None, 50)	1450
dense_1 (Dense)	(None, 10)	510
dense_2 (Dense)	(None, 1)	11
=====		
Total params: 3145 (12.29 KB)		
Trainable params: 3145 (12.29 KB)		
Non-trainable params: 0 (0.00 Byte)		

مطابق متن مقاله یک callback early stopping تعریف می‌شود که بر val_loss کنترل می‌کند و تا 10 مقدار صبر می‌کند و در صورتی که 11 بار val_loss پیشرفتی نداشته باشد بهترین وزن را برمی‌گرداند. مدل با 100 اپیاک و درصد ولیدیشن ۱۰ از دسته داده آموزش برای مدل طبقه بندی و رگرسیون اجرا می‌شود. که مطابق با خواسته شماره دو این مسئله برای مدل رگرسیونی چهار معیار RMSE, MSE, MAE, MAPE باید ارائه شوند که نتایج مدل رگرسیون به شرح زیر است:

MSE: 1240.859497959224,

RMSE: 35.22583566019725,

MAE: 25.886679496185604,

MAPE: 24.776857028177908%

همانطور که در صورت مسئله عنوان شده است نیاز نیست که به مقادیر عددی مقاله برای مدل مذکور نزدیک شویم اما مقایسه ای با این مقادیر نشان می‌دهد که این نتایج با مدل ارائه شده در مقاله در بعضی از معیارهای نزدیک بوده و در بعضی دیگر از معیار های تفاوت قابل توجهی دارد. به طور مثال در معیار MAPE در مقاله مقدار ۲۸.۹۸ ذکر شده است، در حالیکه این مدل به مقدار ۲۴.۷۸ رسیده است. از طرفی

دیگر معیارهای RMSE و MAE در مقاله به ترتیب ۱۷.۹۸ و ۱۴.۱۵ ذکر شده اند در حالیکه در این پیاده سازی این مقادیر عبارتند از ۳۵.۲۳ و ۲۵.۸۹ که تفاوت ویژه ای با مقادیر مقاله دارند.

همچنین در خواسته دیگر سوال نمودار مقادیر واقعی و تخمین زده شده برای RUL ترسیم شده است که مطابق با شکل زیر می باشد:

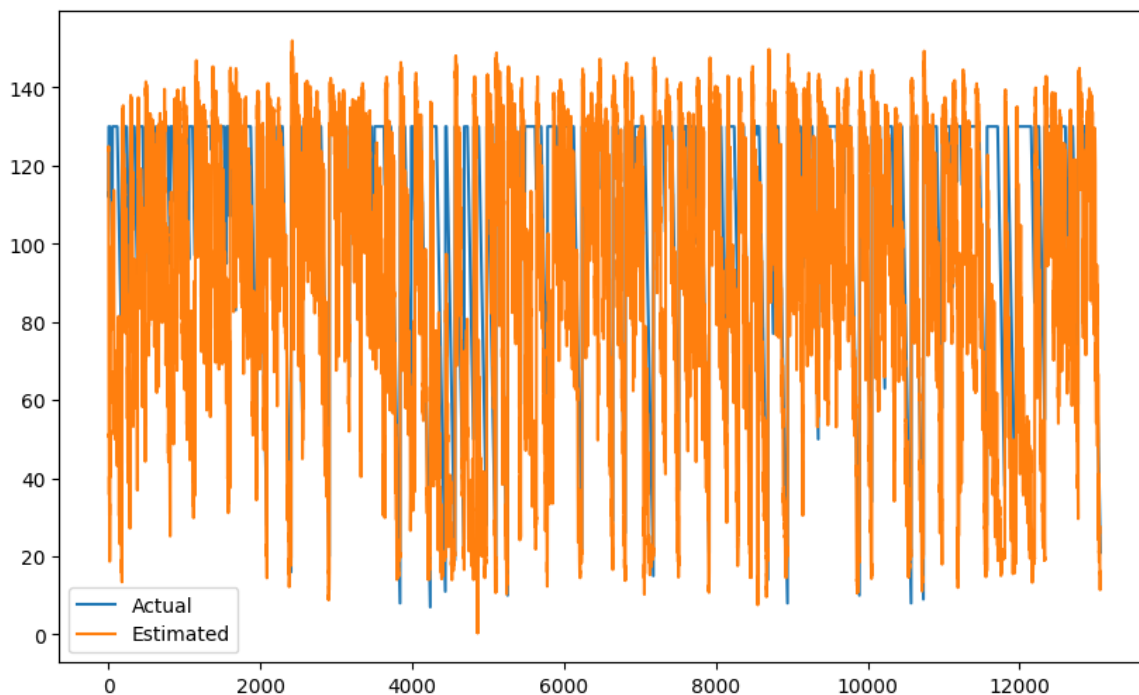


Figure ۲۰- نمودار مقادیر واقعی و پیش بینی شده RUL بوسیله مدل پایه CNN

نتایج مدل طبقه بندی نیز بنا به خواسته قسمت قبل با معیار های precision, recall, f1-score و accuracy میباشد که نتایج این مدل به شرح زیر میباشد :

	precision	recall	f1-score	support
0	0.15	1.00	0.26	331
1	1.00	0.86	0.92	12735
accuracy			0.86	13066
macro avg	0.58	0.93	0.59	13066
weighted avg	0.98	0.86	0.91	13066

با توجه به نتایج موجود، معیار accuracy مقدار قابل قبول ۰.۸۶ را نشان میدهد که این مقدار بسیار کمتر از مدل هیبرید ارائه شده توسط مقاله میباشد و میتوان توجیه مناسب برای این را عدم توانایی مدل های cnn در درک روابط ترتیبی دانست که مدل هیبرید با lstm این ضعف را پوشانده و از ویژگی ها هر دو این مدل ها استفاده میکند تا به معیارهای بالاتری برسد.

همچنین ماتریس در هم‌ریختگی و نمودار ROC نیز مطابق شکل های زیر برای این مدل ترسیم شده‌اند:

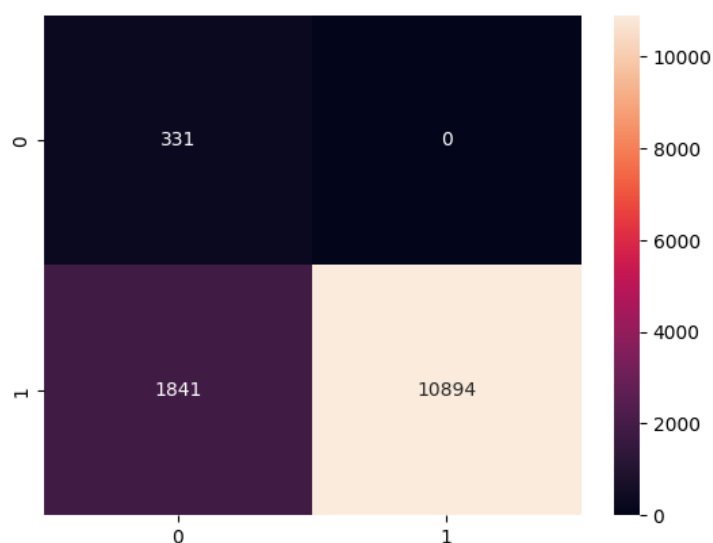


Figure ۲۱- ماتریس در هم‌ریختگی طبقه بندی بوسیله مدل پایه CNN

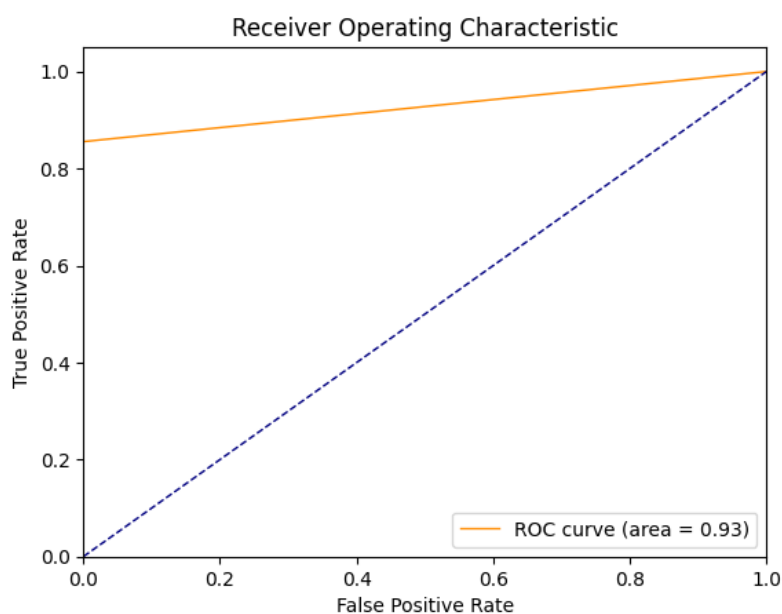


Figure ۲۲- نمودار ROC مدل طبقه بندی مدل پایه CNN

باتوجه به مقادیر جدول و ماتریس در هم‌ریختگی، میتوان متوجه شد باتوجه به درصد ۰.۲۶ برای معیار F-1 برای کلاس صفر، میتوان دریافت که عملکرد مدل در مورد کلاس صفر قابل قبول نبوده هرچند که دقت مدل ۸۶ درصد است اما باید توجه داشت که مدل در شناسایی کلاس صفر دچار نقص است. همچنین مقدار AUC نیز ۰.۹۳ میباشد که در کل عملکرد قابل قبولی را نشان میدهد.

LSTM: مدل

مدل ساخته شده در ابتدا شامل 64 یونیت LSTM با تابع فعال ساز tanh و ابعاد ورودی ۱۷*۳۰ که همان سائز پنجره یا هرکدام از لیست‌های ورودی به مدل می‌باشد. همچنین آرگومان return_sequences برای این ماژول ها در وضعیت true قرار داده شده است. در ادامه دوباره تعداد ۶۴ ماژول LSTM با تابع فعال سازی tanh اما با قرار دادن آرگومان return_sequences در وضعیت false اضافه شد است و در ادامه سه لایه پرسپترون به ترتیب با تعداد نورون ۱۶، ۸ و ۱ به همراه تابع فعال سازی های relu برای دو لایه اول و تابع خطی برای لایه آخر در مدل رگرسیون و تابع sigmoid برای لایه آخر در مدل طبقه بندی قرار داده شده است. Optimizer از نوع RMSprop و تابع loss برای مدل رگرسیون از نوع mse و برای مدل طبقه بندی binary_crossentropy تعریف می‌شود. ساختار لایه‌های مدل مطابق شکل زیر است:

Layer (type)	Output Shape	Param #
lstm_12 (LSTM)	(None, 30, 64)	20992
lstm_13 (LSTM)	(None, 64)	33024
dense_24 (Dense)	(None, 16)	1040
dense_25 (Dense)	(None, 8)	136
dense_26 (Dense)	(None, 1)	9
Total params: 55201 (215.63 KB)		
Trainable params: 55201 (215.63 KB)		
Non-trainable params: 0 (0.00 Byte)		

مطابق متن مقاله یک callback early stopping تعریف می‌شود که بر val_loss کنترل می‌کند و تا 10 مقدار صبر می‌کند و در صورتی که 11 بار val_loss پیشرفتی نداشته باشد بهترین وزن را برمی‌گرداند. مدل با 100 اپیاک و درصد ولیدیشن ۱۰ از دسته داده آموزش برای مدل طبقه بندی و رگرسیون اجرا می‌شود. که مطابق با خواسته شماره دو این مسئله برای مدل رگرسیونی چهار معیار RMSE, MSE, MAE, MAPE باید ارائه شوند که نتایج مدل رگرسیون به شرح زیر است:

MSE: 1455.902166629956,

RMSE: 38.15628606966296,

MAE: 30.104220332959592,

MAPE: 29.685043316986214%

همانطور که در صورت مسئله عنوان شده است نیاز نیست که به مقادیر عددی مقاله برای مدل مذکور نزدیک شویم اما مقایسه ای با این مقادیر نشان میدهد که این نتایج با مدل ارائه شده در مقاله در معیارها تفاوت قابل توجهی دارد. به طور مثال در معیار MAPE در مقاله مقدار ۱۵.۸ ذکر شده است، در حالیکه این مدل به مقدار ۲۹.۶۸ رسیده است. از طرفی دیگر معیارهای RMSE و MAE در مقاله به ترتیب ۱۶.۰۶ و ۱۱.۹۸ ذکر شده اند در حالیکه در این پیاده سازی این مقادیر عبارتند از ۳۸.۱۶ و ۳۰.۱ که تفاوت ویژه ای با مقادیر مقاله دارند.

میانگین خطای مطلق (MAE) که میانگین اختلاف مطلق بین مقادیر برآورد شده و مقدار واقعی را اندازه می گیرد، برای CNN 25.89 و برای LSTM 30.10 است. این بدان معناست که پیش بینی های مدل CNN به طور میانگین به مقادیر واقعی به صورت مطلق نزدیک تر است. همچنین میانگین درصد خطای مطلق (MAPE)، که میانگین خطای مطلق را به صورت درصدی از مقادیر واقعی بیان می کند، برای CNN 24.78 درصد و برای LSTM 29.69 درصد است.

همچنین در خواست دیگر سوال نمودار مقادیر واقعی و تخمین زده شده برای RUL ترسیم شده است که مطابق با شکل زیر می باشد:

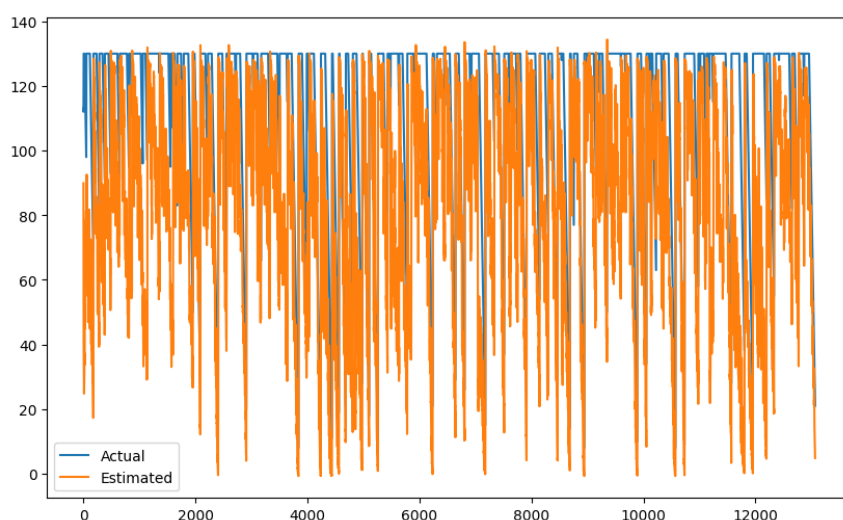


Figure ۲۳- نمودار مقادیر واقعی و پیش بینی شده RUL بوسیله مدل پایه LSTM

نتایج مدل طبقه بندی نیز بنا به خواسته قسمت قبل با معیار های precision, recall, f1-score و accuracy میباشد که نتایج این مدل به شرح زیر میباشد :

	precision	recall	f1-score	support
0	0.27	1.00	0.42	331
1	1.00	0.93	0.96	12735
accuracy			0.93	13066
macro avg	0.63	0.96	0.69	13066
weighted avg	0.98	0.93	0.95	13066

با توجه به نتایج موجود، معیار accuracy مقدار قابل قبول ۰.۹۳ را نشان میدهد که این مقدار عملکرد بهتری را در مقابل مدل CNN رقم زده است که نشان دهنده عملکرد بهتر این ماژول ها در مواجهه با دیتاست های سری زمانی میباشد.

همچنین ماتریس در همریختگی و نمودار ROC نیز مطابق شکل های زیر برای این مدل ترسیم شده اند:

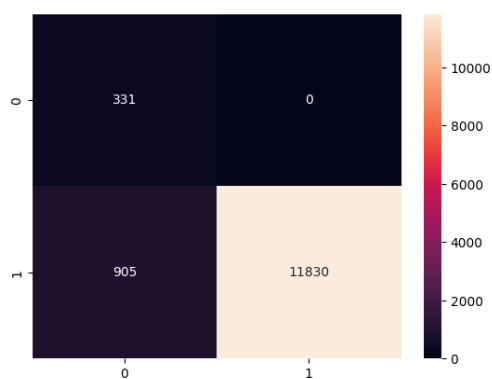


Figure ۲۴- ماتریس در همریختگی طبقه بندی بوسیله مدل پایه LSTM

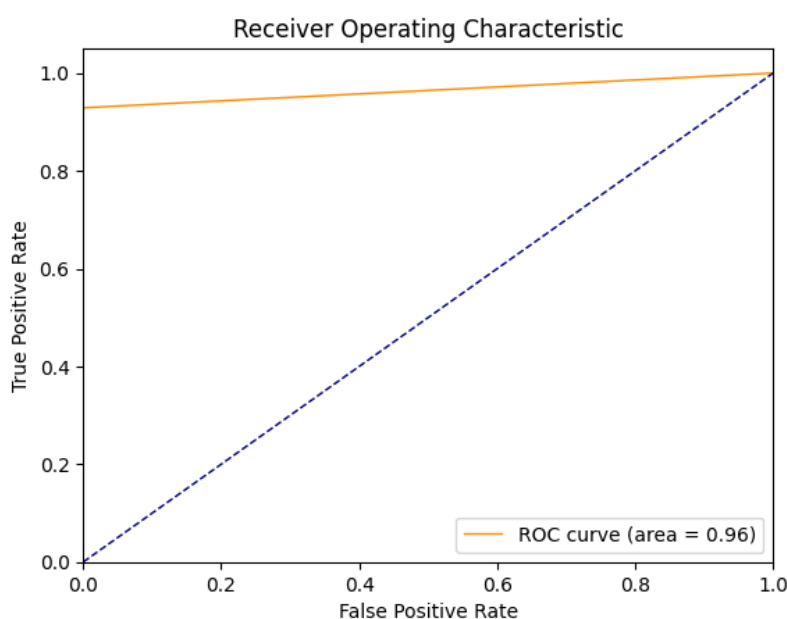


Figure ۲- نمودار ROC مدل طبقه بندی مدل پایه CNN

باتوجه به مقادیر جدول و ماتریس در هم ریختگی، میتوان متوجه شد باتوجه به درصد ۰.۴۲ برای معیار F-1 برای کلاس صفر، میتوان دریافت که عملکرد مدل در مورد کلاس صفر قابل قبول نبوده در حالیکه در مقایسه با عدد ۰.۲۶ در مدل CNN عملکرد بهتری را به نمایش گذاشته است. از طرفی دقت مدل ۹۳ درصد است اما باید توجه داشت که مدل در شناسایی کلاس صفر دچار نقص است. همچنین مقدار AUC نیز ۰.۹۶ میباشد که در کل عملکرد قابل قبولی را نشان میدهد.