Texila International Journal of Public Health
ISSN: 2520-3134
DOI: 10.21522/TIJPH.2013.12.02.Art025

Efficient Multilayer Convolutional Models for Abnormal Heartbeat Signal Detection

Ekta[1]*, Jyoti Sehgal[1], Manoj Kumar[1], Yash Sehgal[2]
[1]*UIET, MDU, Rohtak, India*
[2]*CSE, IIT Mandi, Himachal, Kamad Valley*

Abstract

The escalating ubiquity of heart abnormality extensively, coupled with a multistorey mortality rate, underscores the crucial need for instantaneous and efficacious characteristic measures. Recognizing the censorious nature of this health concern, there's an increasing ultimatum for procedure and machine techniques that can expeditiously and exactly associate these ailments. The reason is to plan a mechanized technique category irregular beat sound prompt to help the surgeon. To the leading of our information, often primary analysis about that employment a single neural organize show sort of 8 diverse sorts of pulse sound signal. In an electrocardiogram (ECG), the electrical action of the heart is recorded and usually spoken to graphically as an arrangement of waves. The ordinary frequency range for an ECG signal is within the run of 0.05 to 150 Hz. The low-frequency components of the ECG signal (0.05 to 1 Hz) constitute the pattern or the slow-changing components of the heart's electrical action. The high-frequency components (1 to 150 Hz) capture the fast changes related to the depolarization and repolarization of the heart's chambers. The recommended show is collated with CNN multilayer perceptron (MLP) in diverse execution assessment lattices. Besides, the results of machine learning (ML) models are moreover examined. Scheduled show accomplished increases classification acc. (multiple layers with dropout) acc. 99.35 (single layer with drop) acc. 98.69 (single layer with no drop) acc. 98.18 (multiple layers with no drop) acc. 99.04 dispute collection of data, which is reliably predominant to its contestant approaches. Representation distributes vital advice to the vascular specialist identifying heart spout ailment.

***Keywords**: CNN Technique, ECG-Based Signal Processing, Machine Learning, Max Pooling, MLP, Spatial Features.*

Introduction

In the 21st century, the integration of innovation and artificial intelligence, including the utilization of machine learning calculations and CNNs, has encouraged and improved our capacity to analyze and anticipate heart disappointment. These instruments complement the ability of human diagnosticians, giving more precise and convenient evaluations [1-3]. A progressive application of Convolutional Neural Systems (CNN) within the domain of healthcare. As cardiovascular well-being takes center organize within the interest of well-being, early discovery of unpredictable heart rhythms gets to be fundamental [4-5]. Leveraging the control of CNN innovation, our Unusual Pulse Locator stands as a cutting-edge arrangement, giving exact and opportune distinguishing proof of sporadic cardiac designs [6-9]. Conventional strategies for recognizing irregular heartbeats regularly depend on complex calculations and broad information

Received: 12.04.2024 ***Accepted: 03.06.2024*** ***Published on: 28.06.2024***
****Corresponding Author: ekta21171@gmail.com***

investigation. In any case, CNNs bring a worldview move by naturally learning unmistakable highlights from electrocardiogram (ECG) information, empowering effective and exact acknowledgement of abnormalities [10].

The convolutional layers of the organism exceed expectations in capturing perplexing designs and varieties, making it a perfect instrument for observing nuanced cardiac inconsistencies [11-13]. This groundbreaking application holds the potential to revolutionize cardiac healthcare, advertising a non-invasive and proactive approach to recognizing unpredictable heart rhythms. The CNN-based Unusual Pulse Locator not as it were upgrades demonstrative exactness but also permits for early intercession, in this manner progressing persistent results and quality of life. Umair B. In this paper, we distinguish the irregular designs of pulse sound prompts from two well-distinguished publicly accessible databases predicated on information surrounding, lessening test estimates, and utilizing diverse ML and DL avenue counting demonstrations [14-16]. The proposed strategy effort viably along with productively analysed heart problems from sound signals gives suitable data to the well-being specialists deciding whether encouraging therapy is needed or not. The clamour was expelled by applying distinctive sifting procedures. Moreover, the inspecting outline of each pulse sound flag was changed over inside a settled rate of distance estimate of 10.0 s, and down sampling procedures utilized extricate foremost overwhelming and discriminative designs [17-18]. The proposed demonstration was connected and accomplished the most noteworthy exactness of 0.9971 and 0.987 on Pascal and PhysioNet datasets separately, which appears strategy distinguishing unusual pulse high audio noteworthy. Future, arranging plan time handles distinguish anomalous pulse designs without utilizing labelled information [19].

A deep system beats category for heart health, different ECG kind representation was inspected escorted by diverse sorts of delta-sigma strategies or comes about were extensively displayed. Perusing ECG gesticulation may be a repetitive assignment indeed for specialists in such areas. except DL procedures demonstrated profound proficiency. crude MIT-BIH data information is profoundly uneven hence began with examining it utilizing destroyed oversampling methods. The other inspecting procedures are consequently utilized along with modified consideration fundamental CNN show, (LENET along with a suggested demonstration, appeared). results prove that the proposed model using Destroyed oversampling strategies gives the leading precision of 98.41%. Additionally, this demonstration has demonstrated to be superior to other existing models that we have investigated amid our inquiry about. A long run opportunity regarding nigher exertion should be via move forward exactness suggested show utilizing hyper-parameter tuning [20]. The abnormal heartbeat detector in heart sound auscultation in virtue of convey accretion in CNN models heart sound auscultation; be that as it may, the utilize of exchange learning continue to exist unexplored. Hence, this consider assesses the execution of such procedures about initial representation, strategy of include ancestors, (VGG16, VGG19, and models, and spectrogram and MFCCs). To entirety up, show considers tried the plausibility of utilize of exchange learn location ECG mumble utilizing PCG recordings. (Utilizing the PASCAL CHSC 2011) database, signals handled examined, outwardly spoken to utilize spectrograms and MFCCs. 3 pre-trained CNN models utilized classified specifically (VGG16, VGG19, and ResNet50), comes about demonstrate utilize exchange learn errand isn't as it were conceivable but gives moderately incredible execution location mumbles, with Spectrogram (ResNet50 pipeline) a classification precision of 87.65%. Whereas the results were essentially moved forward in contrast to precursory performance interminably the identical specifies, an expansive extend concerning upgrade; distinctive intelligent retrieval methods, such as ensemble, can be tried, and a bigger dataset might give an awesome execution in collation via the restricted solitary pre-owned in meditate [21].

An ECG Classification Using Time-Series Augmented Signals, observed f1 esteem starting to demonstrate introductory information set is 0.89, whereas identical show escorted increased data information provided with esteem approx. 0.98 appears enlargement makes a difference in making strides in the expectation precision. Demonstration gives a 0.98 unaccompanied increase whereas with an increase it too gives an f1 score of 0.98, but in this final case, the positioning misfortune and scope mistake esteem is slighter. Applying enlargements to the dataset cannot as it wastes make the demonstration preparation more exact but moreover stability high accuracy. The show comprises six remaining squares suggests there's a purview of over-fitting the information, but the expanded dataset also anticipates over-fitting by manufacture categorization troublesome within the test stage. Show motionless shows tall precision in analogous circumstances. In this manner delineating its calibre to form exceedingly precise forecasts with a precision rate of 99.12% [22]. Manish sh. proposed in this paper automation detection of arrhythmia using optimal orthogonal wavelet filters, through fitting examination of ECG signals and isolating diverse heartbeats conceivable to recognize certain cardiac varieties from standard tallying unmistakable sorts of arrhythmia. In this ponder, perfect wavelet highlights methodology has been made to recognize, recognize and classify unmistakable sorts of arrhythmia ECG signals normally. The system we developed combines 5-level wavelet weakening with a perfect two-channel repeat-specific orthogonal channel bank, followed by highlights extraction, which combines feathery entropy, Renyi entropy, and fractal estimation to collect 18 highlights. These 18 highlights were then used for 10-fold cross-validation classification, with the best results obtained by weighted KNN classifiers. Accuracy of 98%, affectability of 85.33%, and specificity of 98.22% were obtained for set A, which is the database. (without commotion clearing; an accuracy of 98.1%, affectability of 85.63% and specificity of 98.27% of gotten for set by the database with commotion ousted). [23]. The IoT-based Healthcare Monitoring System using Machine Learning, the proposed framework, is ready to conclude that prepared to deliver information which is recognized blocked off warrior pioneer or troopers utilizing handset squad pioneer to main Central utilizing LoRaWAN as the inaccessible transferred development. makes a distinction to screen prosperity parameters warriors, track their position, recognize adjoining and anticipate the utilizing K-Means machine learning calculation. contrasts the trooper to actuate help from an equipped constrained control unit and/or from other person warriors in solidifying circumstances. It'll illustrate to be outstandingly profitable to service qualities throughout the conflict and secure performances since unaccompanied utilized without a course of action control and LoRaWAN. Consequently, this system gives security to troopers. amplified inside long run-in various headings. Whirligig and Accelero-meter can in addition be utilized together for human development affirmation. Blood weight sensor, and electro-dermal can moreover executed in case warrior is in inconvenience. A fitting predominant coordinating calculation utilized makes the system more dependable and imperativeness productive. Omnipresent computing will include all soldier solidifies systems format complete cutting edge. The choice squadron pioneer done statically in this paper though it done effectively inside long utilizing a fitting and compelling cluster-head choice calculation [24]. A profound arrange establish categorization framework identifying heartbeat unusual rhythms low wearable physical sensors. This paper built up compelling classification strategy for heartbeat utilizing critical CNN with a spectrogram input. (ECG signals collected by the ECG sensor through HealthypiV3) is interior the layout of CSV signals and changed over into spectrogram pictures. These pictures were at that point set into the made classified show up based on the Beginning V3 system. It is an optimized CNN classification modeldom fulfil tall exactness in recognizing major ECG issues like (atrial fibrillation, bradycardia, tachycardia, and bundle division square) [25].

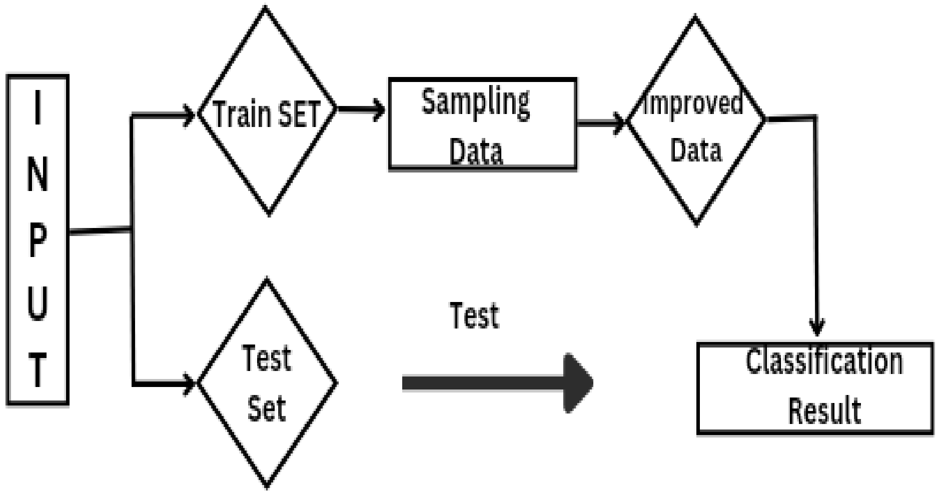


Fig. 1. Classification of ECG Beats using the Proposed Method [3]

The Proposed Methodology

Train Set: The prepare set alludes to the parcel of the dataset utilized to prepare the classification show. Within the setting of ECG classification, this would regularly comprise of an expansive number of ECG recordings with known classifications [31] (e.g., typical cadence, atrial fibrillation, ventricular tachycardia, etc.). These recordings are utilized to prepare the machine learning demonstrate to recognize designs related with each classification.

Test Set: The test set could be an isolated parcel of the dataset that's not utilized amid the preparing stage but is saved to assess the execution of the prepared show [32-33]. It contains ECG recordings comparative to those within the preparing set but is kept isolated to evaluate how well the show generalizes to unused, unseen data.

Sampling Information: Testing information alludes to method of selecting a subset of the available information to represent the whole dataset. Within the setting of ECG classification, this might include arbitrarily selecting a parcel of the ECG recordings from the complete dataset to make the prepare and test sets [34-36].

Made Strides Information: Made strides information seem allude to any improvements or preprocessing strategies connected to the ECG recordings to make strides the quality or value of the data for classification purposes [37].

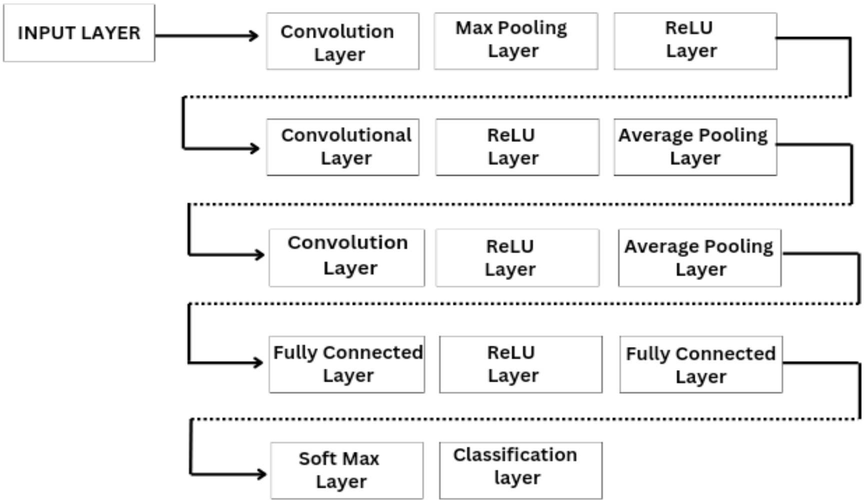


Fig. 2. Typical CNN Classification Segment [26]

Classification Layers of CNN

Convolutional Layer: Convolutional layers apply convolutional filters/kernels to the input picture or highlight maps. These channels identify different designs within the input information, such as edges, surfaces, or shapes. Each channel slides over the input picture, performing element-wise increase and summation, coming about in a highlight outline that highlights the nearness of specific highlights. Different channels are utilized in parallel to capture distinctive highlights [38].

Max Pooling Layer: Max pooling decreases the spatial measurements of the feature maps whereas holding the foremost critical data [39]. It separates the include outline into non-overlapping rectangular districts (pools).

ReLU Layer (Amended Straight Unit): The ReLU enactment work presents non-linearity into the arrange by applying the work f(x)=max (0, x). ReLU sets all negative values within the highlight maps to zero, whereas positive values stay unaltered.

Normal Pooling Layer: Like max pooling, normal pooling diminishes the spatial measurements of the highlight maps. Rather than taking the greatest esteem, normal pooling computes the normal esteem inside each pooling locale [40].

Completely Associated Layer: Totally related layers interface each neuron in one layer to each neuron inside the taking after layer, comparable to conventional manufactured neural networks. The smoothed yield from the past layers serves as the input to the completely associated layer [41]. These layers learn worldwide designs within the highlights extricated by the convolutional layers and are regularly utilized for classification assignments.

Softmax Layer: The SoftMax layer ultimate layer within arrange and commonly utilized for multi-class classification assignments. It applies the SoftMax work to the yield of the past layer, changing over crude scores into probabilities. The SoftMax work guarantees that the probabilities whole up to one, making it appropriate for classification by allotting a likelihood dissemination over the classes.

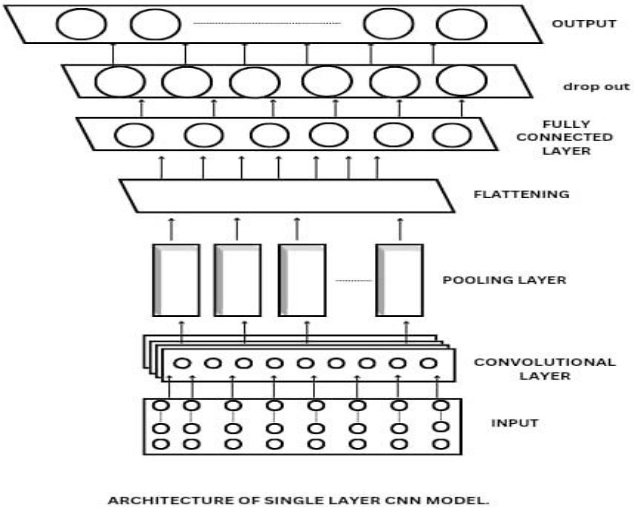


Fig. 3. Single Layer of CNN [26]

Implementation of the Proposed Model

Single Layer Convolutional Neural Network Model

Input Layer: This layer receives the raw input information, regularly pictures in computer vision applications. Each picture is spoken to as a 2D (grayscale) or 3D (color) array of pixel values.

Convolutional Layer: This layer comprises of a set of learnable channels (bits) that slide over the input information, computing dab items to extricate highlights. Each channel captures distinctive designs or highlights show within the input information. After convolution, an actuation work (like ReLU) is regularly connected element-wise to present non-linearity.

Pooling Layer: Pooling layers are utilized to diminish the spatial measurements (width and tallness) of the include maps delivered by the convolutional layer whereas holding important information. Max pooling could be a common pooling operation where the greatest esteem inside a locale (e.g., 2x2 window) is held and others are disposed of.

Straightening: After the convolutional and pooling operations, the highlight maps are regularly smoothed into a one-dimensional vector. This smoothing step changes over the spatial data into a arrange that can be bolstered into the completely associated layer.

Completely Associated Layer (Thick Layer): The straightened yield from the past layers is associated to a completely associated layer. These associations permit organize to memorize complex designs within data. Actuation capacities are connected to the yields of these neurons to present non-linearity. Fully connected Layer: The yield layer produces the ultimate yield of the arrange. For illustration, in classification errands, there could be one neuron per course for parallel classification, or one neuron per lesson together with a softmax enactment work for multi-class classification. In relapse errands, there could be a single neuron for foreseeing a nonstop esteem.

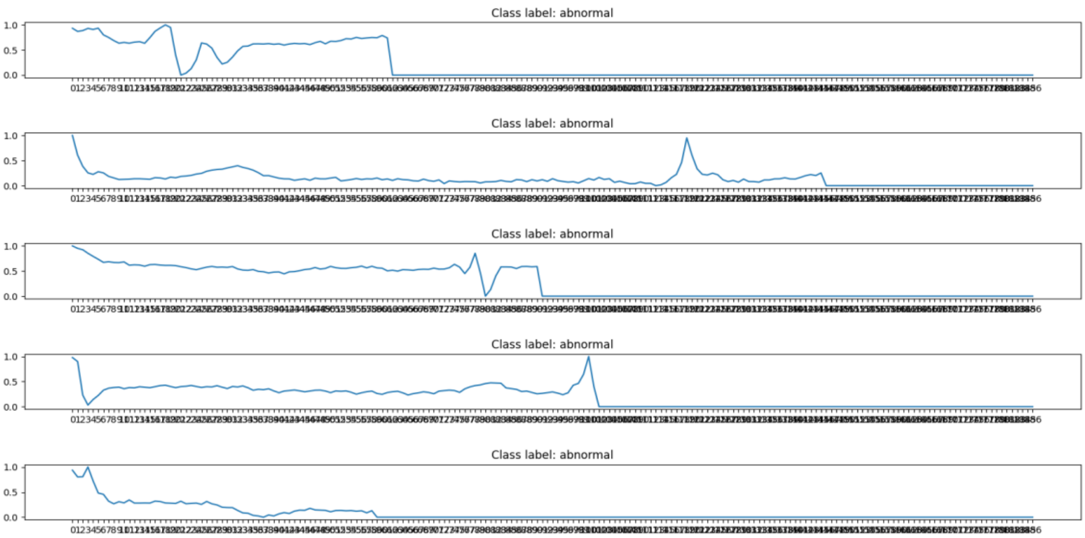
Single layer of CNN without Drop: A single-layer convolutional neural arrange (CNN) without dropout regularly comprises and a completely associated layer. In this setup, the convolutional layer applies a set of learnable channels to the input information, extricating fundamental highlights through a prepare called include mapping. These highlights capture designs and characteristics show within the input information pertinent to the assignment at hand, such as picture classification or protest discovery. Taking after the convolutional layer, a pooling layer is frequently utilized to decrease the spatial measurements of the include maps whereas protecting vital data.

Result Analysis

[22]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from keras.models import Sequential

from keras.layers import Dense ,Conv1D,MaxPool1D,Flatten,Dropout,LSTM
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay,accuracy_score
import tensorflow as tf
```

[23]:
```
data=pd.read_csv("./combined_data.csv")
Y=list((np.array(data["187"])))
class_names=['normal','abnormal']
data.drop(["187"],inplace=True,axis=1)
X=data
```

[24]:
```
fig,ax=plt.subplots(5,1,figsize=(20,10))
plt.subplots_adjust(hspace=1)
for i in range(5):
    ax[i].plot(X.iloc[i, :])
    ax[i].set_title(f'Class label: {class_names[Y[i]]}')
plt.show()
```



In [25]:
```
Y=np.array(Y)
X=np.array(X)
print(X.shape)
X=X.reshape(-1,187,1)

X.shape
Y.shape
```

```
(14552, 187)
```

Out[25]:
```
(14552,)
```

In [26]:
```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=43)
X_train,X_val,Y_train,Y_val=train_test_split(X_train,Y_train,test_size=0.2,random_state=42)
# list(Y_test)
test_lables=Y_test
Y_test=to_categorical(Y_test)
Y_train=to_categorical(Y_train)
Y_val=to_categorical(Y_val)
```

In [27]:
```
model = Sequential()
model.add(Conv1D(128, 5, activation='relu', input_shape=(187, 1)))
model.add(MaxPool1D(pool_size=2))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(2, activation='softmax'))
```

EPOCH	SINGLE LAYER WITHOUT DROP CONVENSION STEP RATE M/S	LOSSES	ACCURACY	VAL_ACCURACY	VAL_ LOSSES
1/100	65ms/step	0.3948e	0.8062	0.8583	0.3323
2/100	58ms/step	0.2774e	0.8837	0.8905	0.2626
3/100	59ms/step	0.2121e	0.9151	0.9279	0.1969
4/100	56ms/step	0.1586e	0.9374	0.9403	0.1662
5/100	56ms/step	0.1340e	0.9503	0.9549	0.1318
6/100	56ms/step	0.1051e	0.9624	0.9309	0.1845
7/100	56ms/step	0.0879e	0.9677	0.9528	0.1265
8/100	56ms/step	0.0688e	0.9779	0.9691	0.1041
9/100	56ms/step	0.0573e	0.9813	0.9678	0.1014
10/100	55ms/step	0.0644e	0.9770	0.9618	0.1113
⋮	⋮	⋮	⋮	⋮	⋮
98/100	62ms/step	2.3553e	1.0000	0.9828	0.1377
99/100	61ms/step	2.1683e	1.0000	0.9837	0.1389
100/100	62ms/step	2.0708e	1.0000	0.9845	0.1386
Accuracy Testing model	4ms/step	0.1298e	0.9818		

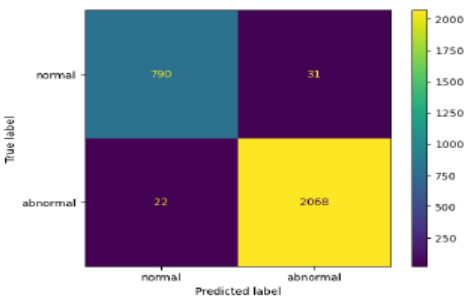
single layer without drop

```
model.summary()
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_3 (Conv1D)           (None, 183, 128)          768

 max_pooling1d_3 (MaxPoolin  (None, 91, 128)           0
 g1D)

 flatten_2 (Flatten)         (None, 11648)             0

 dense_6 (Dense)             (None, 256)               2982144

 dense_7 (Dense)             (None, 128)               32896

 dense_8 (Dense)             (None, 2)                 258

=================================================================
Total params: 3016066 (11.51 MB)
Trainable params: 3016066 (11.51 MB)
Non-trainable params: 0 (0.00 Byte)
```

```
 dense_7 (Dense)             (None, 128)               32896

 dense_8 (Dense)             (None, 2)                 258

=================================================================
Total params: 3016066 (11.51 MB)
Trainable params: 3016066 (11.51 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
In [32]:
predictions=model.predict(X_test)
predicted_labels=np.argmax(predictions,axis=1)
cm=confusion_matrix(test_lables,predicted_labels)
test_accuracy = accuracy_score(test_lables, predicted_labels)
print("Test Accuracy: {:.2%}".format(test_accuracy))
disp=ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=class_names)
disp.plot()
plt.show()
```

```
91/91 [==============================] - 0s 4ms/step
Test Accuracy: 98.18%
```



Single layer with Drop: A single-layer convolutional neural organize (CNN) with dropout consolidates the dropout regularization procedure to upgrade its execution. In this setup, the CNN comprises a convolutional layer, taken after by discretionary pooling, and a completely associated layer.

Result Analysis

In [10]:

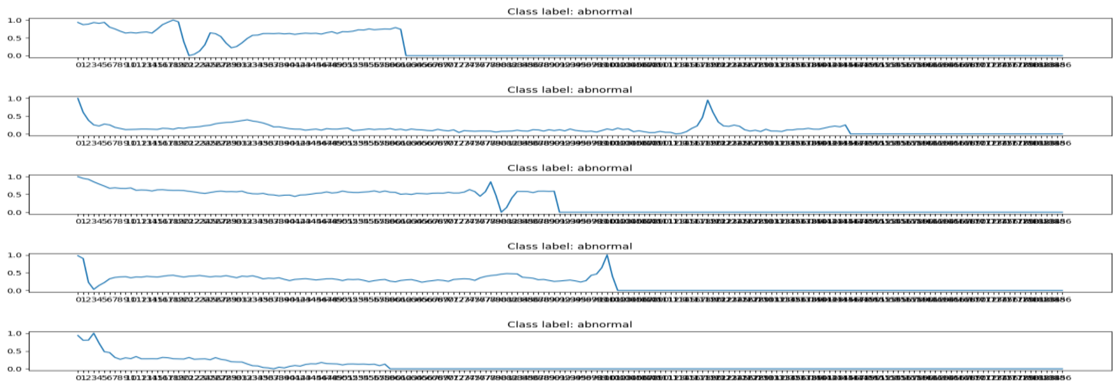
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from keras.models import Sequential

from keras.layers import Dense ,Conv1D,MaxPool1D,Flatten,Dropout,LSTM
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay,accuracy_score
import tensorflow as tf
```

In [11]:

```
data=pd.read_csv("./combined_data.csv")
Y=list((np.array(data["187"])))
class_names=['normal','abnormal']
data.drop(["187"],inplace=True,axis=1)
X=data
```

In [12]:

```
fig,ax=plt.subplots(5,1,figsize=(20,10))
plt.subplots_adjust(hspace=1)
for i in range(5):
    ax[i].plot(X.iloc[i, :])
    ax[i].set_title(f'Class label: {class_names[Y[i]]}')
plt.show()
```



In [13]:

```
Y=np.array(Y)
X=np.array(X)
print(X.shape)
X=X.reshape(-1,187,1)

X.shape
Y.shape
```

```
(14552, 187)
```

Out[13]:

```
(14552,)
```

In [14]:

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=43)
X_train,X_val,Y_train,Y_val=train_test_split(X_train,Y_train,test_size=0.2,random_state=42)
# list(Y_test)
test_lables=Y_test
Y_test=to_categorical(Y_test)
Y_train=to_categorical(Y_train)
Y_val=to_categorical(Y_val)
```

In [15]:

```
model = Sequential()
model.add(Conv1D(128, 5, activation='relu', input_shape=(187, 1)))
model.add(MaxPool1D(pool_size=2))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dense(2, activation='softmax'))
```

In [16]:

```python
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
x=model.fit(X_train,Y_train,validation_data=(X_val,Y_val),epochs=100,batch_size=100)
print("Testing the model")
scores = model.evaluate(X_test, Y_test)
print("Baseline Error: %.2f%%" % (100-scores[1]*100))
```

EPOCH	SINGLE LAYER WITH DROP CONVENSION STEP RATE M/S	LOSSES	ACCURACY	VAL_ ACCURACY	VAL_ LOSSES
1/100	65ms/step	0.4359	0.7859	0.8480	0.3470
2/100	61ms/step	0.3169	0.8630	0.8832	0.2823
3/100	59ms/step	0.2591	0.8924	0.8909	0.2607
4/100	59ms/step	0.2251	0.9085	0.9068	0.2267
5/100	60ms/step	0.1955	0.9213	0.9141	0.2024
6/100	61ms/step	0.1743	0.9301	0.9360	0.1603
7/100	60ms/step	0.1496	0.9422	0.9416	0.1721
8/100	61ms/step	0.1377	0.9472	0.9528	0.1330
9/100	59ms/step	0.1241	0.9531	0.9528	0.1280
10/100	60ms/step	0.1028	0.9635	0.9575	0.1281
⋮	⋮	⋮	⋮	⋮	⋮
98/100	61ms/step	0.0078	0.9968	0.9871	0.0992
99/100	62ms/step	0.0124	0.9963	0.9867	0.1015
100/100	62ms/step	0.0127	0.9953	0.9858	0.1007
Accuracy Testing model	4ms/step	0.0835	0.9869		

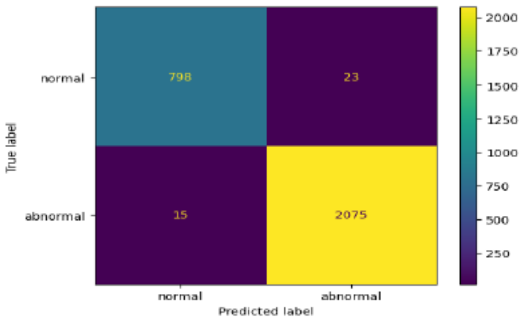
[19]: model.summary()

single layer with drop

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape
Param #
=================================================================
 conv1d_2 (Conv1D)           (None, 183, 1
28)         768

 max_pooling1d_2 (MaxPoolin  (None, 91, 12
8)          0
 g1D)

 flatten_1 (Flatten)         (None, 11648)
0

 dense_3 (Dense)             (None, 256)
2982144

 dropout (Dropout)           (None, 256)
0

 dense_4 (Dense)             (None, 128)
32896

 dense_5 (Dense)             (None, 2)
258

=================================================================
Total params: 3016066 (11.51 MB)
Trainable params: 3016066 (11.51 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
predictions=model.predict(X_test)
predicted_labels=np.argmax(predictions,axis=1)
cm=confusion_matrix(test_lables,predicted_labels)
test_accuracy = accuracy_score(test_lables, predicted_labels)
print("Test Accuracy: {:.2%}".format(test_accuracy))
disp=ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=class_names)
disp.plot()
plt.show()
```

```
91/91 [==============================] - 0s 4ms/step
Test Accuracy: 98.69%
```



Multi-Layer of CNN Model

Multi-Layer without Drops: A multi-layer convolutional neural organize (CNN) without dropout may be a significant learning designing comprising of various stacked convolutional layers, pooling layers, and totally related layers. In this setup, each convolutional layer applies learnable channels to the input data, removing continuously one of a kind and complex highlight as the data passes through the organize. These highlights capture dynamic representations of the input data, allowing the organize to memorize affluent plans and structures. Pooling layers are routinely scattered between convolutional layers to decrease the spatial estimations of the highlight maps though securing crucial information. This spatial diminishment makes a contrast control the computational complexity of the orchestrate and expects overfitting by centering on the preeminent germane highlights.

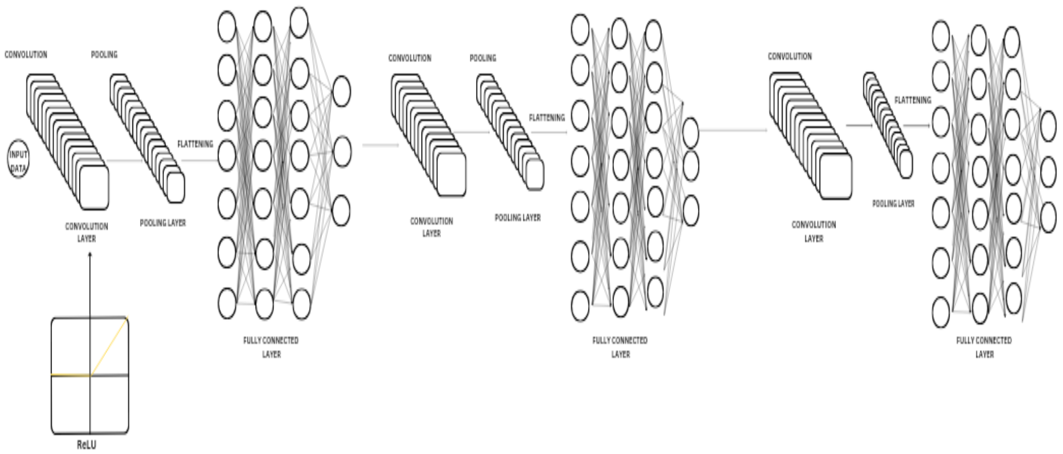


Fig. 4. Multi-Layer of CNN [2]

Result Analysis

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from keras.models import Sequential

from keras.layers import Dense ,Conv1D,MaxPool1D,Flatten,Dropout,LSTM
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay,accuracy_score
import tensorflow as tf
```

```
WARNING:tensorflow:From c:\Users\yashs\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\losses.p
y:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softma
x_cross_entropy instead.
```

In [2]:
```
data=pd.read_csv("./combined_data.csv")
Y=list((np.array(data["187"])))
class_names=['normal','abnormal']
data.drop(["187"],inplace=True,axis=1)
X=data
```

In [3]:
```
fig,ax=plt.subplots(5,1,figsize=(20,10))
plt.subplots_adjust(hspace=1)
for i in range(5):
    ax[i].plot(X.iloc[i, :])
    ax[i].set_title(f'Class label: {class_names[Y[i]]}')
plt.show()
```

In [4]:
```
Y=np.array(Y)
X=np.array(X)
print(X.shape)
X=X.reshape(-1,187,1)

X.shape
Y.shape
```

```
(14552, 187)
```

Out[4]:
```
(14552,)
```

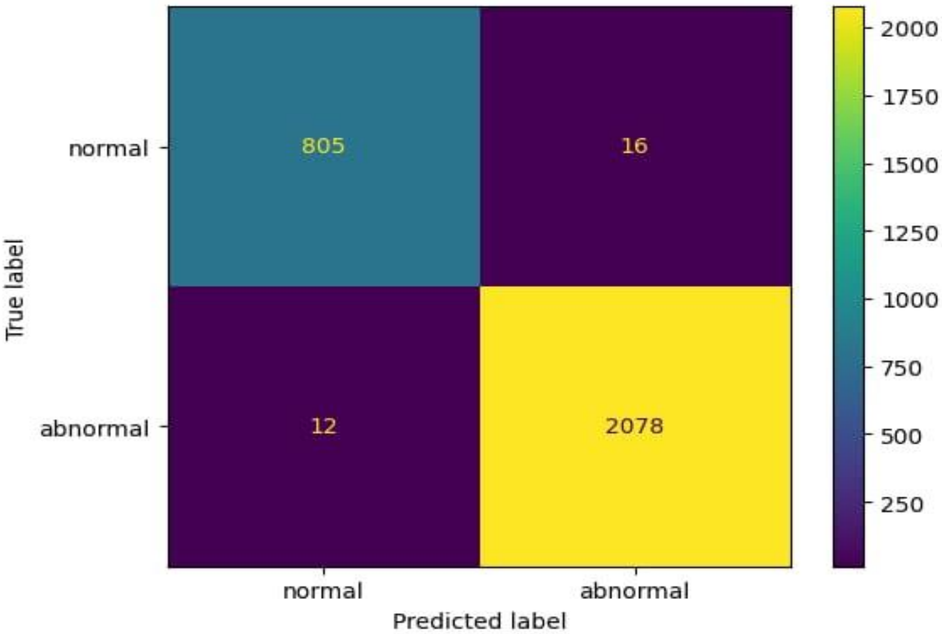
In [5]:
```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=43)
X_train,X_val,Y_train,Y_val=train_test_split(X_train,Y_train,test_size=0.2,random_state=42)
# list(Y_test)
test_lables=Y_test
Y_test=to_categorical(Y_test)
Y_train=to_categorical(Y_train)
Y_val=to_categorical(Y_val)
```

In [6]:
```
model = Sequential()
model.add(Conv1D(128, 5, activation='relu', input_shape=(187, 1)))
model.add(MaxPool1D(pool_size=2))
model.add(Conv1D(64, 3, activation='relu'))
model.add(MaxPool1D(pool_size=2))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(2, activation='softmax'))
```

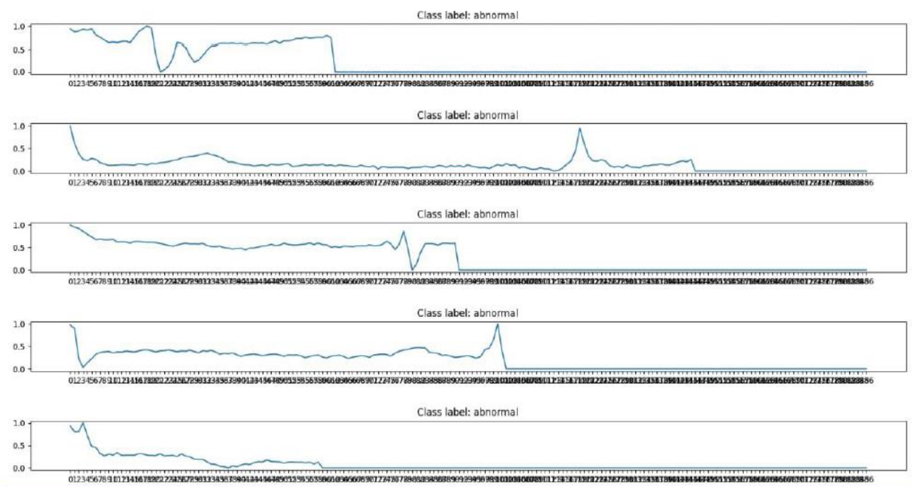
In [7]:
```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
x=model.fit(X_train,Y_train,validation_data=(X_val,Y_val),epochs=100,batch_size=100)
print("Testing the model")
scores = model.evaluate(X_test, Y_test)
print("Baseline Error: %.2f%%" % (100-scores[1]*100))
```

In [9]:
```
predictions=model.predict(X_test)
predicted_labels=np.argmax(predictions,axis=1)
cm=confusion_matrix(test_lables,predicted_labels)
test_accuracy = accuracy_score(test_lables, predicted_labels)
print("Test Accuracy: {:.2%}".format(test_accuracy))
disp=ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=class_names)
disp.plot()
plt.show()
```

```
91/91 [==============================] - 0s 3ms/step
Test Accuracy: 99.04%
```



EPOCH	MULTI-LAYER WITHOUT DROP CONVENSION STEP RATE M/S	LOSSES	ACCURACY	VAL_ ACCURACY	VAL_ LOSSES
1/100	35ms/step	0.4407	0.7789	0.8429	0.3529
2/100	29ms/step	0.3172	0.8618	0.8914	0.2698
3/100	30ms/step	0.2400	0.9020	0.9020	0.2172
4/100	31ms/step	0.1605	0.9395	0.9395	0.1582
5/100	29ms/step	0.1199	0.9561	0.9561	0.1457
6/100	29ms/step	0.0910	0.9663	0.9663	0.0987
7/100	28ms/step	0.0784	0.9715	0.9715	0.1096
8/100	29ms/step	0.0554	0.9821	0.9821	0.1187
9/100	28ms/step	0.0463	0.9870	0.9848	0.0842
10/100	28ms/step	0.0380	0.9819	0.9870	0.1218
⋮	⋮	⋮	⋮	⋮	⋮
98/100	28ms/step	1.4128e	1.0000	0.9850	0.1276
99/100	28ms/step	1.3383e	1.0000	0.9854	0.1288
100/100	28ms/step	1.2598e	1.0000	0.9850	0.1296
Accuracy Testing model	3ms/step	0.0637	0.9904		



Multi-layer with Drop: A multi-layer convolutional neural arranges (CNN) with dropout could be an effective profound learning engineering that joins dropout regularization to progress its execution and generalization capacity. This organize comprises of numerous stacked convolutional layers, pooling layers, and completely associated layers, comparable to the setup of a multi-layer CNN without dropout. In this setup, each convolutional layer applies learnable channels to the input information, extricating progressively theoretical and complex highlights as the information passes through the arrange. Dropout is at that point connected after one or more convolutional layers, which includes haphazardly deactivating a division of the neurons amid preparing.

Result Analysis

In [8]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from keras.models import Sequential

from keras.layers import Dense ,Conv1D,MaxPool1D,Flatten,Dropout,LSTM
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay,accuracy_score
import tensorflow as tf
```

In [9]:
```
data=pd.read_csv("./combined_data.csv")
Y=list((np.array(data["187"])))
class_names=['normal','abnormal']
data.drop(["187"],inplace=True,axis=1)
X=data
```

In [10]:
```
fig,ax=plt.subplots(5,1,figsize=(20,10))
plt.subplots_adjust(hspace=1)
for i in range(5):
    ax[i].plot(X.iloc[i, :])
    ax[i].set_title(f'Class label: {class_names[Y[i]]}')
plt.show()
```

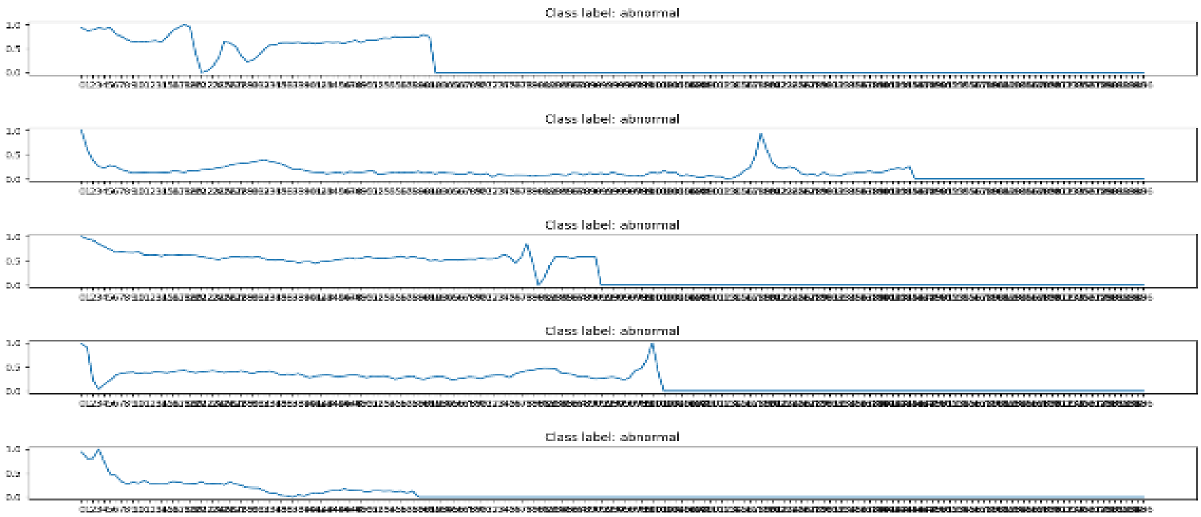
In [11]:
```
Y=np.array(Y)
X=np.array(X)
print(X.shape)
X=X.reshape(-1,187,1)

X.shape
Y.shape
```

```
(14552, 187)
```

Out[11]:
```
(14552,)
```

In [12]:
```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=43)
X_train,X_val,Y_train,Y_val=train_test_split(X_train,Y_train,test_size=0.2,random_state=42)
# list(Y_test)
test_lables=Y_test
Y_test=to_categorical(Y_test)
Y_train=to_categorical(Y_train)
Y_val=to_categorical(Y_val)
```

In [13]:
```
model = Sequential()
model.add(Conv1D(128, 5, activation='relu', input_shape=(187, 1)))
model.add(MaxPool1D(pool_size=2))
model.add(Conv1D(64, 3, activation='relu'))
model.add(MaxPool1D(pool_size=2))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
```



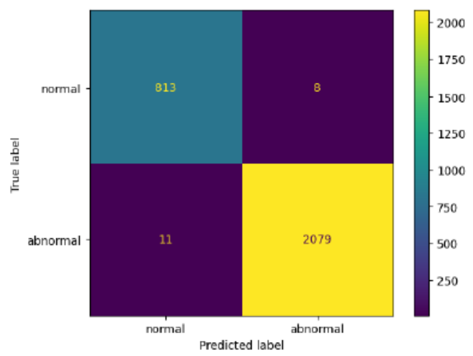
In [14]:
```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
x=model.fit(X_train,Y_train,validation_data=(X_val,Y_val),epochs=100,batch_size=100)
print("Testing the model")
scores = model.evaluate(X_test, Y_test)
print("Baseline Error: %.2f%%" % (100-scores[1]*100))
```

EPOCH	MULTI-LAYER WITH DROP CONVENSION STEP RATE M/S	LOSSES	ACCURACY	VAL_ACCURACY	VAL_ LOSSES
1/100	30ms/step	0.4355	0.7865	0.8094	0.3854
2/100	29ms/step	0.3170	0.8637	0.8789	0.2891
3/100	28ms/step	0.2619	0.8887	0.8905	0.2675
4/100	29ms/step	0.2202	0.9108	0.9043	0.2281
5/100	30ms/step	0.1827	0.9298	0.9450	0.1428
6/100	29ms/step	0.1428	0.9474	0.9506	0.1314
7/100	29ms/step	0.1145	0.9573	0.9463	0.1443
8/100	37ms/step	0.1181	0.9572	0.9678	0.1015
9/100	46ms/step	0.0863	0.9703	0.9605	0.1138
10/100	42ms/step	0.0744	0.9736	0.9712	0.0894
⋮	⋮	⋮	⋮	⋮	⋮
98/100	46ms/step	0.0050	0.9983	0.9910	0.0616
99/100	29ms/step	0.0017	0.9997	0.9897	0.0578
100/100	32ms/step	0.0030	0.9989	0.9910	0.0599
Accuracy Testing model	5ms/step	0.0454	0.9935		

Testing the model
91/91 [==============================] - 0s 5ms/step - loss: 0.0454 - accuracy: 0.9935
Baseline Error: 0.65%

In [15]:
```
predictions=model.predict(X_test)
predicted_labels=np.argmax(predictions,axis=1)
cm=confusion_matrix(test_lables,predicted_labels)
test_accuracy = accuracy_score(test_lables, predicted_labels)
print("Test Accuracy: {:.2%}".format(test_accuracy))
disp=ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=class_names)
disp.plot()
plt.show()
```

```
91/91 [==============================] - 1s 4ms/step
Test Accuracy: 99.35%
```



Conclusion

To recognize the possible hole and un-addressed issues inside the ECG heart-beat arrhythmia assessment issue, the previous considers are coordinated into a taxonomical system. Different methodologies proposed inside the composing connected with bunching and classification of ECG heart-beat arrhythmia have been kept an eye on. Such coordinated composing concentrate on grants investigators for the conspicuous verification of fissure and the ask about issues ignored up to this point. Openings and ignored issues have been summed up so that the experts might research approaches for keeping an eye on them in this work.

References

[1] Shivnarayan Patidar, Ram Bilas Pachori, 2014, Classification of Cardiac Sound Signals Using Constrained Tunable-Q Wavelet Transform", *Expert Systems with Applications*, Volume 41, Issue 16, 7161-7170, https://doi.org/10.1016/j.eswa.2014.05.052

[2] Manab K., Das corresponding author and Samit Ari, 2014 , Patient-specific ECG Beat Classification Technique, National Library of Medicine, Doi: 10.1049/htl.2014.0072, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4611171/

[3] lvarado, A. S., Lakshminarayan, C., Principe, J. C, 2012, Time-Based Compression and Classification of Heartbeats, *IEEE Transactions on Biomedical Engineering* 59, 1641–1648, Doi: 10.1109/TBME.2013.2238191, https://ieeexplore.ieee.org/document/6409262?denied=

[4] de Chazal, P., Reilly, R. B, 2006, A Patient-Adapting Heartbeat Classifier Using ECG Morphology and Heartbeat Interval Features, *IEEE Transactions on Biomedical Engineering* 53, 2535–2543, Doi: 10.1109/TBME.2006.883802, https://ieeexplore.ieee.org/document/4015601

[5] Christov, I., Jekova, I., Bortolan, G, 2005, Premature Ventricular Contraction Classification by the k^{th} Nearest-Neighbours Rule, *Physiological Measurement* 26, 123, Doi: 10.1088/0967-3334/26/1/011, https://iopscience.iop.org/article/10.1088/0967-3334/26/1/011/meta

[6] Hassaan Malik, Umair Bashir, Adnan Ahmad, 2022, Multi-Classification Neural Network Model for Detection of Abnormal Heartbeat Audio signal,

Volume 4, 100048, https://doi.org/10.1016/j.bea.2022.100048
[7] Velagapudi Swapna Sindhu, Kavuri Jaya Lakshmi, Ameya Sanjanita Tangellamudi, K. Ghousiya Begum, 2023, A Novel Deep Neural Network Heartbeats Classifier for Heart Health Monitoring, Volume 4, 1-10, https://doi.org/10.1016/j.ijin.2022.11.001
[8] Omair Rashed Abdulwareth Almanifi, Ahmad Fakhri Ab Nasir, Mohd Azraai Mohd Razman, Rabiu Muazu Musa, Anwar, P. P., AbdulMajeed, 2022, Heartbeat Murmurs Detection in Phonocardiogram *Recordings via Transfer Learning, Journal* 61, 10995–11002, https://doi.org/10.1016/j.aej.2022.04.031
[9] Pratik Kanania, Mamta Padoleb, 2020, ECG Heartbeat Arrhythmia Classification using Time-Series Augmented Signals and Deep Learning Approach, *Procedia Computer Science* 171, 524–531, https://doi.org/10.1016/j.procs.2020.04.056
[10] Manish Sharma, Ru-San Tan, U. Rajendra Acharya, 2019, Automated Heartbeat Classification and Detection of Arrhythmia using Optimal Orthogonal Wavelet Filters, *Informatics in Medicine Unlocked* 16, 100221, https://doi.org/10.1016/j.imu.2019.100221
[11] Aashay Gondalia a, Dhruv Dixit b, Shubham Parashar c, Vijayanand Raghava d, Animesh Sengupta e, Vergin Raja Sarobin, 2018, IoT-based Healthcare Monitoring System for War Soldiers using Machine Learning, *Procedia Computer Science*, Volume 133, 1005-1013, https://doi.org/10.1016/j.procs.2018.07.075
[12] Edward, B., Panganiban, a., Arnold, C., Paglinawan, b., Wen Yaw Chung c., Gilbert Lance S., Paa, d., 2021, ECG Diagnostic Support System (EDSS): A Deep Learning Neural Network Based Classification System for Detecting ECG Abnormal Rhythms from a Low-Powered Wearable Biosensors, *Sensing and Bio-Sensing Research,* Volume 31, 100398, https://doi.org/10.1016/j.sbsr.2021.100398
[13] Jonas Chromik, Lukas Pirl, Jossekin Beilharz, Bert Arnrich, Andreas Polze, 2021, Certainty in QRS Detection with Artificial Neural Networks, IEEE Conference, Volume 68, 102628, https://doi.org/10.1016/j.bspc.2021.102628
[14] Zhongwei Jiang, Samjin Choi, 2006, A Cardiac Sound Characteristic Waveform Method for in-home Heart Disorder Monitoring with Electric Stethoscopes, Volume 31, Issue 2, 286-298, https://doi.org/10.1016/j.eswa.2005.09.02510
[15] D. Kumar, P. Carvalho, M. Antunes, P. Gil, J. Henriques, L. Eugenio, 2006, A New Algorithm for Detection of S1 and S2 Heart Sounds", IEEE Conference, DOI:10.1109/ICASSP.1660559, https://www.semanticscholar.org/paper/A-New-Algorithm-for-Detection-of-S1-and-S2-Heart-Kumar-Carvalho/120916432fa36d0c2c0d1d1a685a682eae42080d
[16] De Lannoy, G., Francois, D., Delbeke, J., Verleysen, M, 2012, Weighted Conditional Random Fields for Supervised Interpatient Heartbeat Classification, *IEEE Transactions on Biomedical Engineering*, Doi: 10.1109/TBME.2011.2171037, https://pubmed.ncbi.nlm.nih.gov/21990327/
[17] Guler, I., Ubeylı, E. D, 2005, ECG Beat Classifier Designed by Combined Neural Network Model, Pattern Recognition 38(2):199-208, Doi:10.1016/S0031-3203(04)00276-6, https://www.researchgate.net/publication/271584317_ECG_beat_classifier_designed_by_combined_neural_network_model
[18] Das, M. K., Ari, S., 2014, Patient-Specific ECG Beat Classification Technique, Healthcare Technology Letters, PMID: 26609386 PMCID: PMC4611171 DOI: 10.1049/htl.2014.0072 https://pubmed.ncbi.nlm.nih.gov/26609386/
[19] Hu, Y. H., Palreddy, S., Tompkins, W. J, 1997, A Patient-Adaptable ECG Beat Classifier Using A Mixture of Experts Approach, *IEEE Transactions on Biomedical Engineering* 44, 891–900, DOI 10.1109/10.623058, https://ieeexplore.ieee.org/document/623058
[20] Huang, H., Liu, J., Zhu, Q., Wang, R., Hu, G, 2014, Detection of Inter-Patient Left and Right Bundle Branch Block Heartbeats in ECG using Ensemble Classifiers, *Biomedical Engineering*

Online 13, 72, https://doi.org/10.1186/1475-925X-13-72

[21] Huang, H., Liu, J., Zhu, Q., Wang, R., Hu, G., 2014, A New Hierarchical Method for Inter-Patient Heartbeat Classification using Random Projections and rr intervals, *Biomedical Engineering Online* 13, 90, https://doi.org/10.1186/1475-925X-13-90

[22] Ince, T., Kiranyaz, S., Gabbouj, M., 2009, A Generic and Robust System for Automated Patient-Specific Classification of ECG signals, *IEEE Transactions on Biomedical Engineering* 56, 1415–1426, DOI: 10.1109/TBME.2009.2013934, https://ieeexplore.ieee.org/document/4776456

[23] Jacobs, R. A., 1995, Methods for Combining Experts' Probability Assessments", *Neural Computation* 7, 867–888, https://direct.mit.edu/neco/article-abstract/7/5/867/5891/Methods-For-Combining-Experts-Probability?redirectedFrom=fulltext

[24] Liao, T. W., 2005, Clustering of Time Series Data—A Survey, Pattern Recognition 38, 1857–1874, https://doi.org/10.1016/j.patcog.2005.01.025

[25] Llamedo, M., Mart́ınez, J. P., 2011, Heartbeat Classification Using Feature Selection Driven by Database Generalization Criteria, *IEEE Transactions on Biomedical Engineering* 58, 616–625.DOI: 10.1109/TBME.2010.2068048, https://ieeexplore.ieee.org/document/5551181

[26] Salem, A. B. M., Revett, K., El-Dahshan, E., 2009. Machine Learning in Electrocardiogram Diagnosis. International Multiconference on Computer Science and Information Technology, https://www.proceedings2009.imcsit.org/pliks/10.pdf