Model 0 - the model from Neutron Star paper overfitting , 8 epochs, 2.3 million params
Model 1 - 8 epochs, 100% acc training set, 75% validation set acc.

```
def model_creation():
  n_timesteps, n_features = 16384, 1
  input_shape=(n_timesteps,n_features)
  model=Sequential()

  model.add(Conv1D(filters=32, kernel_size=16, activation='relu',
input_shape=input_shape))
  model.add(MaxPooling1D(pool_size=4))

  model.add(Conv1D(64, kernel_size=8, activation='relu'))
  model.add(MaxPooling1D(pool_size=4))

  model.add(Conv1D(128, kernel_size=8, activation='relu'))
  #model.add(Conv1D(16, kernel_size=16, activation='relu'))
  model.add(MaxPooling1D(pool_size=4))

  model.add(Conv1D(256, kernel_size=8, activation='relu'))
  #model.add(Conv1D(32, kernel_size=16, activation='relu'))
  model.add(MaxPooling1D(pool_size=4))

  model.add(Flatten())

  model.add(Dense(128, activation='relu'))
  model.add(Dropout(0.5))

  model.add(Dense(64, activation='relu'))
  model.add(Dropout(0.5))

  model.add(Dense(1, activation='sigmoid'))

  model.compile(optimizer= Adam(learning_rate=0.002, beta_1=0.9,
beta_2=0.999, epsilon=1e-08),
                loss='binary_crossentropy',
                metrics=['binary_accuracy', 'accuracy'])
  model.summary()
  return model
```

Model 2- 10 epochs - 97% on training and 96% on validation, ckpt_2

```
def model_creation():
  n_timesteps, n_features = 16384, 1
  input_shape=(n_timesteps,n_features)
  model=Sequential()

  model.add(Conv1D(filters=16, kernel_size=16, activation='relu',
input_shape=input_shape))
  model.add(MaxPooling1D(pool_size=8))

  model.add(Conv1D(32, kernel_size=8, activation='relu'))
  model.add(MaxPooling1D(pool_size=8))

  model.add(Conv1D(64, kernel_size=8, activation='relu'))
  #model.add(Conv1D(16, kernel_size=16, activation='relu'))
  model.add(MaxPooling1D(pool_size=4))

  model.add(Conv1D(128, kernel_size=8, activation='relu'))
  #model.add(Conv1D(32, kernel_size=16, activation='relu'))
  model.add(MaxPooling1D(pool_size=4))

  model.add(Flatten())

  model.add(Dense(128, activation='relu'))
  model.add(Dropout(0.3))

  model.add(Dense(64, activation='relu'))
  model.add(Dropout(0.3))

  model.add(Dense(1, activation='sigmoid'))

  model.compile(optimizer= Adam(learning_rate=0.002, beta_1=0.9,
beta_2=0.999, epsilon=1e-08),
                loss='binary_crossentropy',
                metrics=['binary_accuracy', 'accuracy'])
  model.summary()
  return model
```

Model 3 - same above model for 15 epochs, ckpt_3, better than model2. 98.18% train acc, 96% validation acc.

Model 4- 97% training & val set acc., better than model 2.

Model 5 - 1.06 params, 14 epochs, 98.53% training acc, 98.22 validation acc.,

```
def model_creation():
  n_timesteps, n_features = 16384, 1
  input_shape=(n_timesteps,n_features)
  model=Sequential()

  model.add(Conv1D(filters=8, kernel_size=8, activation='relu',
input_shape=input_shape))
  # model.add(Conv1D(8, kernel_size=8, activation='relu'))
  model.add(MaxPooling1D(pool_size=4))

  model.add(Conv1D(16, kernel_size=8, activation='relu'))
  # model.add(Conv1D(16, kernel_size=8, activation='relu'))
  model.add(MaxPooling1D(pool_size=4))

  model.add(Conv1D(32, kernel_size=8, activation='relu'))
  # model.add(Conv1D(64, kernel_size=8, activation='relu'))
  model.add(MaxPooling1D(pool_size=4))

  # model.add(Conv1D(64, kernel_size=8, activation='relu'))
  # model.add(Conv1D(128, kernel_size=8, activation='relu'))
  # model.add(MaxPooling1D(pool_size=4))

  model.add(Flatten())
  # model.add(GlobalAveragePooling1D(data_format='channels_last'))
  model.add(Dense(128, activation='relu'))
  model.add(Dropout(0.3))

  model.add(Dense(128, activation='relu'))
  model.add(Dropout(0.3))

  model.add(Dense(64, activation='relu'))
  model.add(Dropout(0.3))

  model.add(Dense(1, activation='sigmoid'))
```

```
  model.compile(optimizer= Adam(learning_rate=0.002, beta_1=0.9,
beta_2=0.999, epsilon=1e-08),
                loss='binary_crossentropy',
                metrics=['binary_accuracy', 'accuracy'])
  model.summary()
  return model
```