Window size:1024
Stride: 1024
Use all data and all channels
No scale on raw signal. Shuffle and split data to 20% for test. Shuffle and split the rest to 80% for training and 20% for validation.

```
model = keras.models.Sequential()
model.add(layers.InputLayer(input_shape=X[:,:,:].shape[1:]))
model.add(layers.Bidirectional(layers.LSTM(32,return_sequences=True,
                                                    recurrent_regularizer=reg)))
model.add(layers.Conv1D(filters=32,
kernel_size=kernel_size,strides=1,padding='same',kernel_regularizer=reg))
model.add(layers.BatchNormalization(momentum=0.8))
model.add(layers.ELU())
model.add(layers.AveragePooling1D(2))
model.add(layers.Dropout(0.5))
model.add(layers.Conv1D(filters=16,
kernel_size=kernel_size,strides=1,padding='same',kernel_regularizer=reg))
model.add(layers.BatchNormalization(momentum=0.8))
model.add(layers.ELU())
model.add(layers.AveragePooling1D(2))
model.add(layers.Dropout(0.5))
model.add(layers.Conv1D(filters=8,
kernel_size=kernel_size,strides=1,padding='same',kernel_regularizer=reg))
model.add(layers.BatchNormalization(momentum=0.8))
model.add(layers.ELU())
model.add(layers.AveragePooling1D(2))
model.add(layers.Dropout(0.5))
model.add(layers.Conv1D(filters=4,
kernel_size=kernel_size,strides=1,padding='same',kernel_regularizer=reg))
model.add(layers.BatchNormalization(momentum=0.8))
model.add(layers.ELU())
model.add(layers.GlobalAveragePooling1D())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(3,activation='softmax',kernel_regularizer=reg))
```

```
kernel_size=9
reg=regularizers.l2(1e-4)
class weights = {0:1, 1:20}
```

Train (acc 0.993)

	Predicted 0	Predicted others
Actual 0	2620	18
Actual others	0	340

Validation (acc 0970)

	Predicted 0	Predicted others
Actual 0	664	16
Actual others	6	59

Test 20% (acc 0.949)

	Predicted 0	Predicted others
Actual 0	868	42
Actual others	10	106