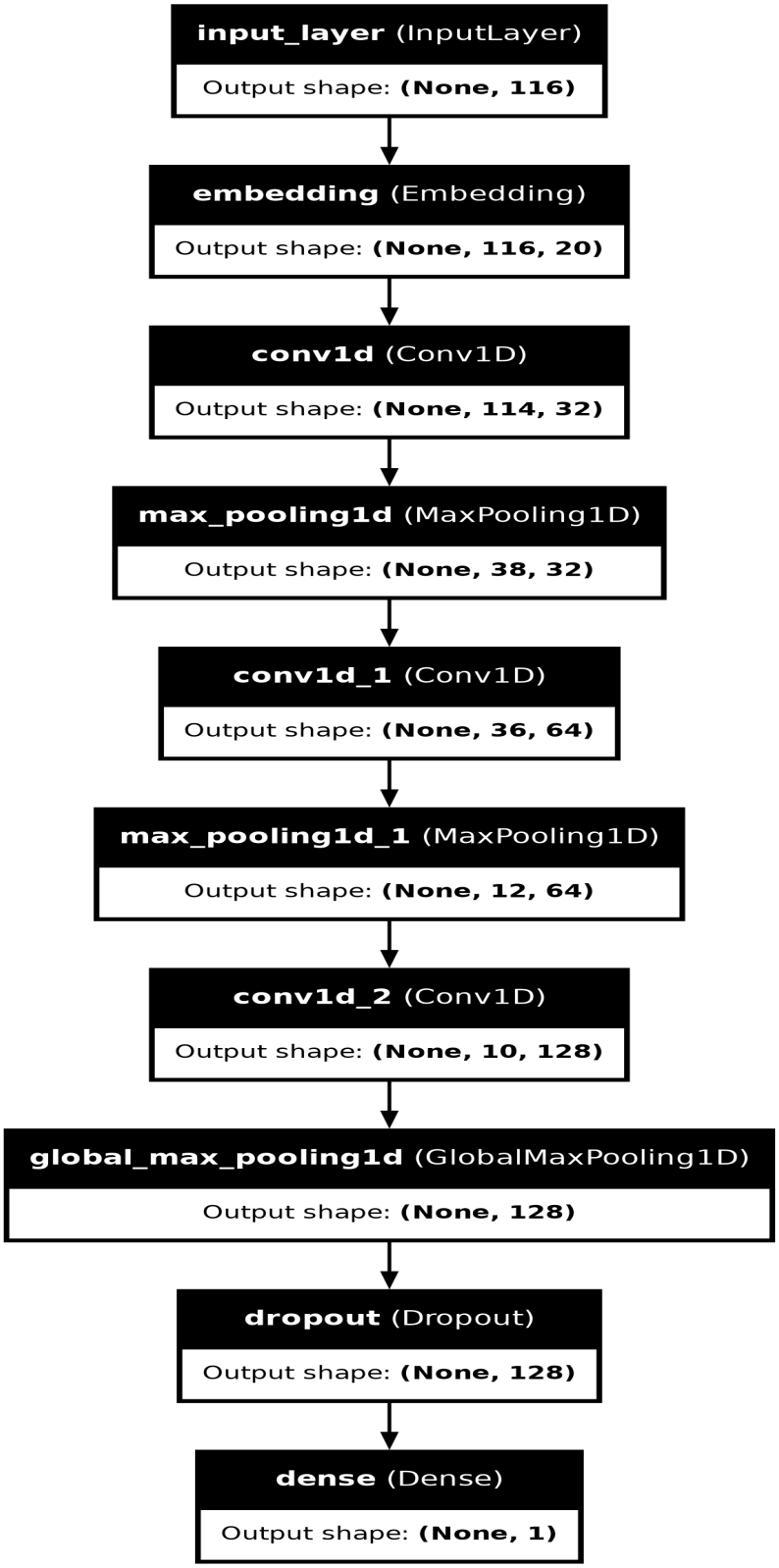
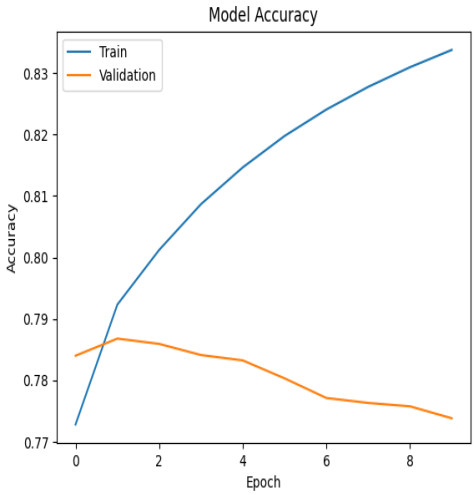
Model 1:

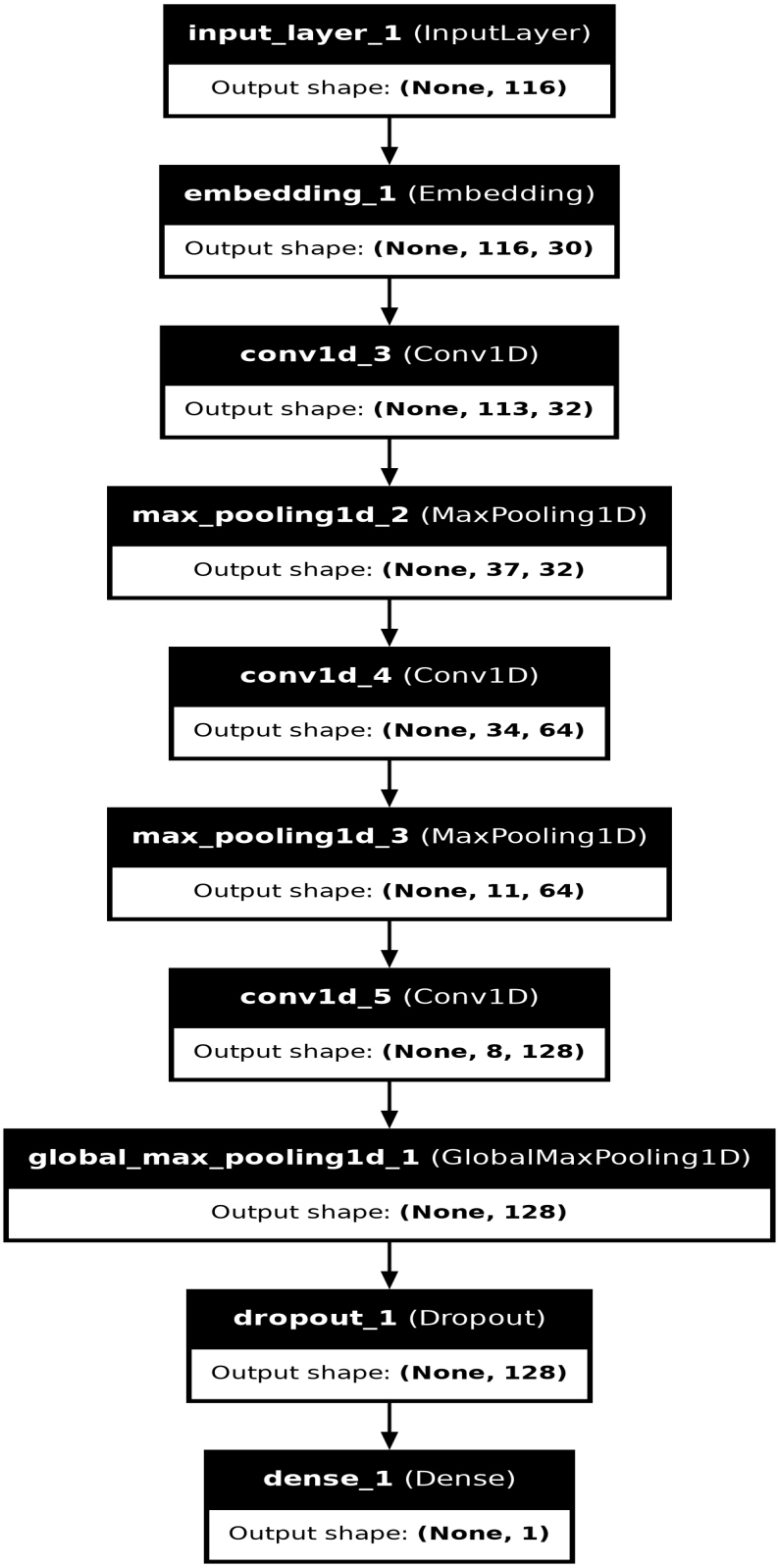
Parameters: Embedding dimension = 20, Window = 3, 1- Conv1D no. filters = 32, 2- Conv1D no. filters = 64 3- Conv1D no. filters = 128

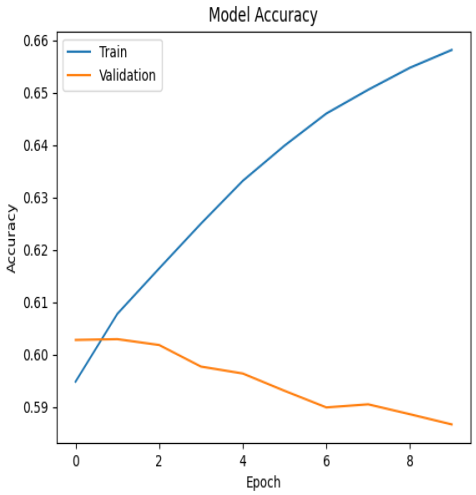

input_layer (InputLayer)
Output shape: (None, 116)
embedding (Embedding)
Output shape: (None, 116, 20)
conv1d (Conv1D)
Output shape: (None, 114, 32)
max_pooling1d (MaxPooling1D)
Output shape: (None, 38, 32)
conv1d_1 (Conv1D)
Output shape: (None, 36, 64)
max_pooling1d_1 (MaxPooling1D)
Output shape: (None, 12, 64)
conv1d_2 (Conv1D)
Output shape: (None, 10, 128)
global_max_pooling1d (GlobalMaxPooling1D)
Output shape: (None, 128)
dropout (Dropout)
Output shape: (None, 128)
dense (Dense)
Output shape: (None, 1)



Model 2:

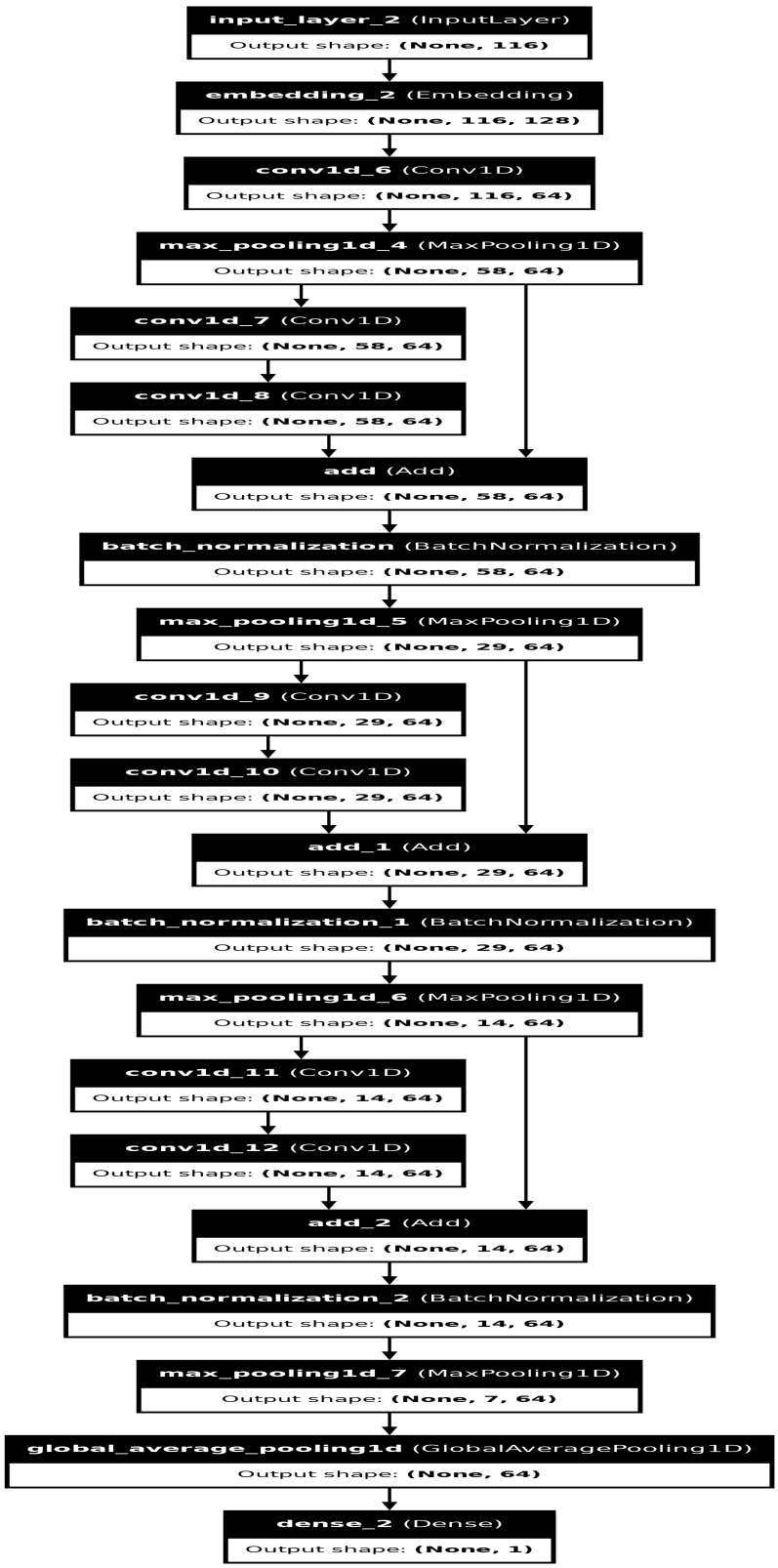
Parameters: Embedding dimension = 30, Window = 4, 1- Conv1D no. filters = 32, 2- Conv1D no. filters = 64 3- Conv1D no. filters = 128

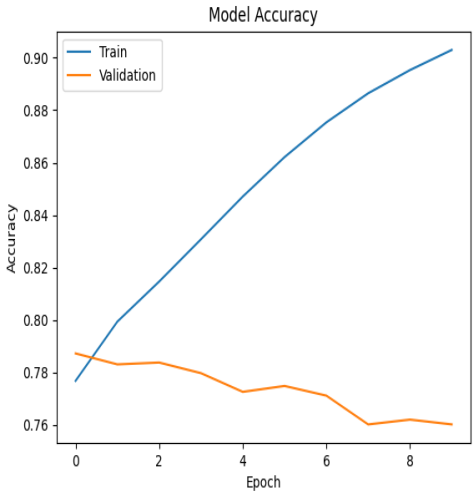

input_layer_1 (InputLayer)
Output shape: (None, 116)
embedding_1 (Embedding)
Output shape: (None, 116, 30)
conv1d_3 (Conv1D)
Output shape: (None, 113, 32)
max_pooling1d_2 (MaxPooling1D)
Output shape: (None, 37, 32)
conv1d_4 (Conv1D)
Output shape: (None, 34, 64)
max_pooling1d_3 (MaxPooling1D)
Output shape: (None, 11, 64)
conv1d_5 (Conv1D)
Output shape: (None, 8, 128)
global_max_pooling1d_1 (GlobalMaxPooling1D)
Output shape: (None, 128)
dropout_1 (Dropout)
Output shape: (None, 128)
dense_1 (Dense)
Output shape: (None, 1)



Model 3:

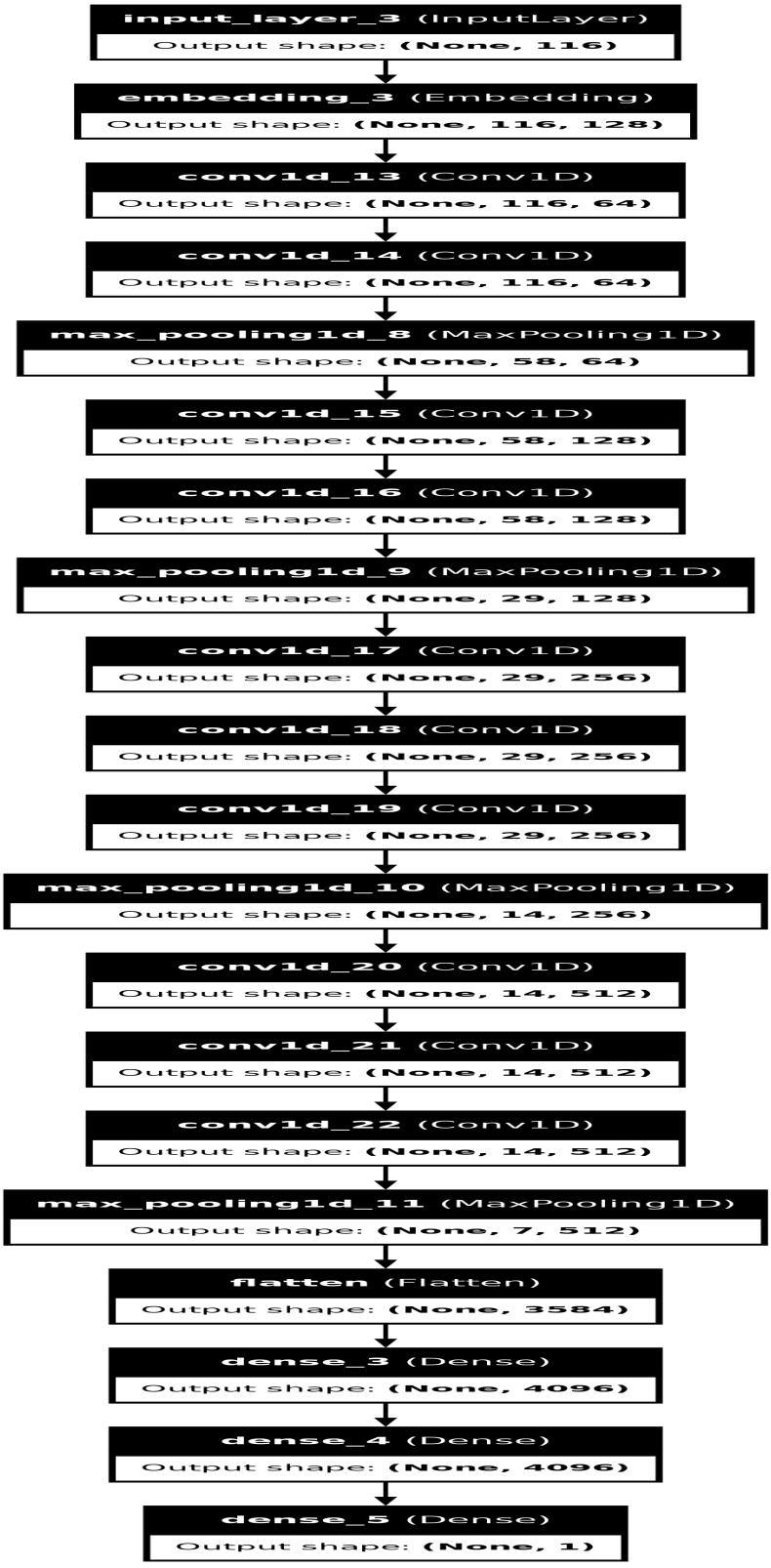
Architecture: {'filters': 64, 'kernel_size': 3, 'num_blocks': 3}

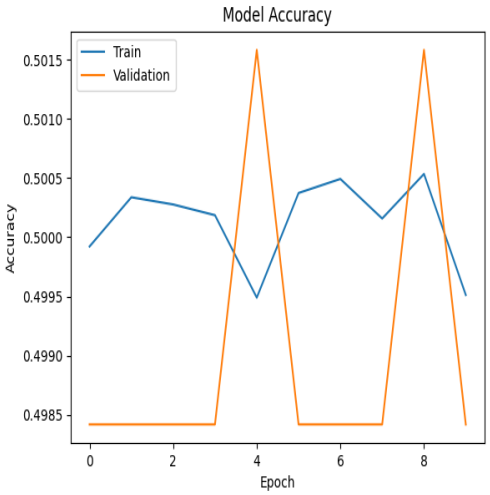

input_layer_2 (InputLayer)
Output shape: (None, 116)
embedding_2 (Embedding)
Output shape: (None, 116, 128)
conv1d_6 (Conv1D)
Output shape: (None, 116, 64)
max_pooling1d_4 (MaxPooling1D)
Output shape: (None, 58, 64)
conv1d_7 (Conv1D)
Output shape: (None, 58, 64)
conv1d_8 (Conv1D)
Output shape: (None, 58, 64)
add (Add)
Output shape: (None, 58, 64)
batch_normalization (BatchNormalization)
Output shape: (None, 58, 64)
max_pooling1d_5 (MaxPooling1D)
Output shape: (None, 29, 64)
conv1d_9 (Conv1D)
Output shape: (None, 29, 64)
conv1d_10 (Conv1D)
Output shape: (None, 29, 64)
add_1 (Add)
Output shape: (None, 29, 64)
batch_normalization_1 (BatchNormalization)
Output shape: (None, 29, 64)
max_pooling1d_6 (MaxPooling1D)
Output shape: (None, 14, 64)
conv1d_11 (Conv1D)
Output shape: (None, 14, 64)
conv1d_12 (Conv1D)
Output shape: (None, 14, 64)
add_2 (Add)
Output shape: (None, 14, 64)
batch_normalization_2 (BatchNormalization)
Output shape: (None, 14, 64)
max_pooling1d_7 (MaxPooling1D)
Output shape: (None, 7, 64)
global_average_pooling1d (GlobalAveragePooling1D)
Output shape: (None, 64)
dense_2 (Dense)
Output shape: (None, 1)



Model 4:

Architecture: VGG for text


input_layer_3 (InputLayer)
Output shape: (None, 116)
embedding_3 (Embedding)
Output shape: (None, 116, 128)
conv1d_13 (Conv1D)
Output shape: (None, 116, 64)
conv1d_14 (Conv1D)
Output shape: (None, 116, 64)
max_pooling1d_8 (MaxPooling1D)
Output shape: (None, 58, 64)
conv1d_15 (Conv1D)
Output shape: (None, 58, 128)
conv1d_16 (Conv1D)
Output shape: (None, 58, 128)
max_pooling1d_9 (MaxPooling1D)
Output shape: (None, 29, 128)
conv1d_17 (Conv1D)
Output shape: (None, 29, 256)
conv1d_18 (Conv1D)
Output shape: (None, 29, 256)
conv1d_19 (Conv1D)
Output shape: (None, 29, 256)
max_pooling1d_10 (MaxPooling1D)
Output shape: (None, 14, 256)
conv1d_20 (Conv1D)
Output shape: (None, 14, 512)
conv1d_21 (Conv1D)
Output shape: (None, 14, 512)
conv1d_22 (Conv1D)
Output shape: (None, 14, 512)
max_pooling1d_11 (MaxPooling1D)
Output shape: (None, 7, 512)
flatten (Flatten)
Output shape: (None, 3584)
dense_3 (Dense)
Output shape: (None, 4096)
dense_4 (Dense)
Output shape: (None, 4096)
dense_5 (Dense)
Output shape: (None, 1)



Conclusion:
According to the accuracies of each architecture, it appears that the first one one outperforms the others.