Findings

Using hashed payload is not worth it. Information is lost. There is a significant drop in accuracy.

LSTM splitt using rounded-down cut TRAINING

model = keras.Sequential([  
 keras.layers.Dense(65),  
 keras.layers.LSTM(512),  
 keras.layers.Dense(2, activation=tf.nn.softmax)  
])  
model.compile(optimizer=keras.optimizers.RMSprop(lr=0.05),  
 loss='sparse\_categorical\_crossentropy',  
 metrics=['accuracy'])  
  
model.fit(trainingdata[0], trainingdata[1], epochs=100)

900/900 [==============================] - 41s 46ms/sample - loss: 0.4077 - acc: 0.9267

Good: 739, Bad: 161

Moving window, size 100, using Event Name. 0.86 acc

50: 84798/84798 [==============================] - 132s 2ms/sample - loss: 1.0346 - acc: 0.8325

Good: 70579, Bad: 14219

Adagrad 50

84798/84798 [==============================] - 127s 2ms/sample - loss: 0.2932 - acc: 0.8829

Good: 70579, Bad: 14219

Using only 16 LSTM = 78865/78865 [==============================] - 127s 2ms/sample - loss: 0.4267 - acc: 0.8431

Good: 66596, Bad: 12269

Adagrad 100, 200 epochs

78865/78865 [==============================] - 248s 3ms/sample - loss: 0.2963 - acc: 0.8829

Good: 66596, Bad: 12269

Supersmall

78865/78865 [==============================] - 63s 801us/sample - loss: 0.4351 - acc: 0.8442

Good: 66596, Bad: 12269

Good 28 bad 72

LSTMSplitted: 0.48 acc, 0.67 when 50/50

LSTMSplitted2: 0.64 acc, 0.78 when 50/50

LSTMSplitted3: 0.36 acc, 0.5 when 50/50

LSTMSplittedTemporal: 0.28 acc, 0.51 when 50/50

LSTMSplittedTemporal\_50: 0.27 acc, 50 when 50/50