Model	Number	Max. validation set accuracy
DNN model = Sequential() model.add(layers.Dense(64, activation='relu', input_dim=input_size)) model.add(layers.Dense(64, activation='relu')) model.add(layers.Dense(3, activation='softmax')) model.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=['accuracy'])	1	.5141
LSTM model = Sequential() model.add(Embedding(18, 32)) model.add(LSTM(32)) model.add(Dense(1, activation='sigmoid')) # model.compile(loss='mae', optimizer='adam')	2	0.28
DNN model = Sequential() model.add(layers.Dense(64, activation='relu', input_dim=input_size)) model.add(layers.Dense(64, activation='relu')) model.add(layers.Dense(3, activation='softmax')) model.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=['accuracy'])	3	.5507
LSTM model = Sequential() model.add(Embedding(64, 128)) model.add(LSTM(64)) model.add(Dense(1, activation='softmax')) # model.compile(loss='mae', optimizer='adam')	4	0.46
LSTM model = Sequential() model.add(Embedding(8, 64)) model.add(LSTM(8)) model.add(Dropout(0.5)) model.add(Dense(1, activation='relu')) # model.compile(loss='mae', optimizer='adam') model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])	5	0.61

LSTM
model = Sequential()
  model.add(Embedding(14, 32))
  model.add(LSTM(13))
  model.add(Dropout(0.1))
  model.add(Dense(1, activation='relu'))
# model.compile(loss='mae', optimizer='adam') 	
 	
  model.compile(loss='mae', optimizer='adam',
metrics=['accuracy'])
	6	0.2701

LSTM
np.random.seed(24)
  tf.random.set_seed(69)
 	
model = Sequential()
  model.add(Embedding(4, 32))
  model.add(LSTM(32))
  model.add(Dropout(0.1))
  model.add(Dense(1, activation='sigmoid'))
# model.compile(loss='mae', optimizer='adam') 	
 	
  model.compile(loss='binary_crossentropy',
optimizer='rmsprop', metrics=['accuracy'])
	7	0.2279

LSTM
model = Sequential()
  model.add(Embedding(4210, 14))
  model.add(SpatialDropout1D(0.2))
  model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
  model.add(Dense(1, activation='sigmoid'))
  model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
  # model.compile(loss='mae', optimizer='adam')
	8	0.22

DNN
model = Sequential()
  model.add(layers.Dense(100, activation='tanh',
input_dim=input_size))
  model.add(layers.Dense(100, activation='tanh',
input_dim=input_size))
  model.add(layers.Dense(3, activation='softmax'))
  model.compile(loss='categorical_crossentropy',
optimizer='Adam', metrics=['accuracy'])
	9	0.5402

DNN np.random.seed(24) tf.random.set_seed(69) model = Sequential() model.add(layers.Dense(100, activation='tanh', input_dim=input_size)) model.add(layers.Dense(100, activation='relu', input_dim=input_size)) model.add(layers.Dense(3, activation='softmax')) model.compile(loss='categorical_crossentropy', optimizer='Adam', metrics=['accuracy'])	10	0.51
DNN model = Sequential() model.add(Dense(8, activation='relu', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])	11	.6298
DNN model = Sequential() model.add(Dense(8, activation='relu', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])	12	.6535
DNN model = Sequential() model.add(Dense(16, activation='relu', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])	13	.6455
DNN model = Sequential() model.add(Dense(12, activation='relu', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])	14	.6561

DNN model = Sequential() model.add(Dense(12, activation='softmax', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])	15	.6311
DNN model = Sequential() model.add(Dense(12, activation='softplus', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])	16	.6508
DNN model = Sequential() model.add(Dense(12, activation='tanh', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])	17	.6520
DNN model = Sequential() model.add(Dense(12, activation='selu', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])	18	.6535
DNN model = Sequential() model.add(Dense(12, activation='elu', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])	19	.6535
DNN model = Sequential()	20	.6561

model.add(Dense(12, activation='exponential', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])		
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='SGD', metrics=['accuracy'])	21	.6430
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='RMSprop', metrics=['accuracy'])	22	.6469
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Adadelta', metrics=['accuracy'])	23	.6509
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Adagrad', metrics=['accuracy'])	24	.6469
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size)) model.add(Dense(1, activation='sigmoid'))	25	.6455

model.compile(loss='binary_crossentropy', optimizer='Adamax', metrics=['accuracy'])		
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	26	.6588
DNN model = Sequential() model.add(Dense(15, activation='exponential', input_dim = input_size)) model.add(Dense(12, activation='exponential')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	27	.6522
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size)) model.add(Dense(15, activation='exponential')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	28	.6495
DNN model.add(Dense(12, activation='exponential', input_dim = input_size)) model.add(Dense(15, activation='relu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	29	.6548
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='random_normal')) model.add(Dense(1, activation='sigmoid'))	30	.6522

model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])		
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size)) model.add(Dense(12, activation='relu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	31	.6588
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='random_uniform')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	32	.6548
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='truncated_normal')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	33	.6522
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='ones')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	34	.4677
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='zeros')) model.add(Dense(1, activation='sigmoid'))	35	.6535

model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])		
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='glorot_normal')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	36	.6495
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='glorot_uniform')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	37	.6561
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='identity')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	38	.6456
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='orthogonal')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	39	.6667
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='constant')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	40	.6522

DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='orthogonal', kernel_regularizer=regularizers.l1(1e-5))) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	41	.6456
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='orthogonal', kernel_regularizer=regularizers.l1(1e-2))) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	42	.6324
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='orthogonal', kernel_regularizer=regularizers.l1(1e-3))) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	43	.6430
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='orthogonal', kernel_regularizer=regularizers.l1(1e-7))) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	44	.6535
DNN model = Sequential() model.add(Dense(12, activation='exponential', input_dim = input_size, kernel_initializer='orthogonal', kernel_regularizer=regularizers.l2(1e-5))) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy',	45	.6535

optimizer='Nadam', metrics=['accuracy'])		

RNN
model = Sequential()
  model.add(Embedding(input_size, 5))
  model.add(SimpleRNN(5))
  model.add(Dense(1, activation='sigmoid'))
 	
  model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])
	46	.5613

RNN
model = Sequential()
  model.add(Embedding(input_size, 12))
  model.add(SimpleRNN(12))
  model.add(Dense(1, activation='sigmoid'))
 	
  model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])
	47	.5545

GRU
model = Sequential()
  model.add(Embedding(17, 32))
  model.add(GRU(units=2))
  model.add(Dense(1, activation='sigmoid'))
 	
  model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])
	48	.5599

GRU
model = Sequential()
  model.add(Embedding(17, 32))
  model.add(GRU(units=5))
  model.add(Dense(1, activation='sigmoid'))
 	
  model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])
	49	.5573

GRU
model = Sequential()
  model.add(Embedding(17, 32))
  model.add(GRU(units=10))
  model.add(Dense(1, activation='sigmoid'))
 	
  model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])
	50	.5583

RNN	51	.5547

model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=5)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])		
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=10)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	52	.5534
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20)) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	53	.5573
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=10, activation='relu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	54	.5573
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=10, activation='sigmoid')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	55	.5583
RNN model = Sequential() model.add(Embedding(17, 32))	56	.5583

model.add(SimpleRNN(units=10, activation='softmax')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])		
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=10, activation='softplus')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	57	.5583
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=10, activation='softsign')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	58	.5543
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=10, activation='selu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	59	.5560
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=10, activation='elu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	60	.5543
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=10, activation='exponential')) model.add(Dense(1, activation='sigmoid'))	61	.0000

model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])		
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='relu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	62	.5573
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='sigmoid')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	63	.5560
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='softmax')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	64	.5583
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='softplus')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	65	.5623
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='softsign')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy',	66	.5583

optimizer='Nadam', metrics=['accuracy'])		
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='selu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	67	.5583
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='elu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	68	.5573
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='exponential')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	69	.0000
GRU model = Sequential() model.add(Embedding(17, 32)) model.add(GRU(units=10, activation='relu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	70	.5583
GRU model = Sequential() model.add(Embedding(17, 32)) model.add(GRU(units=10, activation='sigmoid')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	71	.5583
GRU	72	.5583

model = Sequential() model.add(Embedding(17, 32)) model.add(GRU(units=10, activation='softmax')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])		
GRU model = Sequential() model.add(Embedding(17, 32)) model.add(GRU(units=10, activation='softplus')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	73	.5583
GRU model = Sequential() model.add(Embedding(17, 32)) model.add(GRU(units=10, activation='softsign')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	74	.5583
GRU model = Sequential() model.add(Embedding(17, 32)) model.add(GRU(units=10, activation='selu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	75	.5573
GRU model = Sequential() model.add(Embedding(17, 32)) model.add(GRU(units=10, activation='elu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])	76	.5587
GRU model = Sequential() model.add(Embedding(17, 32))	77	.5583

model.add(GRU(units=10, activation='exponential')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Nadam', metrics=['accuracy'])		
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='softplus')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='SGD', metrics=['accuracy'])	78	.0000
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='softplus')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='RMSprop', metrics=['accuracy'])	79	.5626
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='softplus')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Adam', metrics=['accuracy'])	80	.5613
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='softplus')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Adadelta', metrics=['accuracy'])	81	.5613
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='softplus')) model.add(Dense(1, activation='sigmoid'))	82	.5639

model.compile(loss='binary_crossentropy', optimizer='Adagrad', metrics=['accuracy'])		
RNN model = Sequential() model.add(Embedding(17, 32)) model.add(SimpleRNN(units=20, activation='softplus')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Adamax', metrics=['accuracy'])	83	.5626
GRU model = Sequential() model = Sequential() model.add(Embedding(17, 32)) model.add(GRU(units=10, activation='elu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='SGD', metrics=['accuracy'])	84	.5323
GRU model = Sequential() model = Sequential() model.add(Embedding(17, 32)) model.add(GRU(units=10, activation='elu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='RMSprop', metrics=['accuracy'])	85	.5573
GRU model = Sequential() model = Sequential() model.add(Embedding(17, 32)) model.add(GRU(units=10, activation='elu')) model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy', optimizer='Adadelta', metrics=['accuracy'])	86	.5539