ДОДАТОК А

Програмний код реалізації методу прогнозування цін цифрового активу

```
def shuffle_in_unison(a, b):
    assert len(a) == len(b)
    shuffled_a = np.empty(a.shape, dtype=a.dtype)
    shuffled_b = np.empty(b.shape, dtype=b.dtype)
    permutation = np.random.permutation(len(a))
    for old_index, new_index in enumerate(permutation):
        shuffled_a[new_index] = a[old_index]
        shuffled_b[new_index] = b[old_index]
    return shuffled_a, shuffled_b
def create_Xt_Yt(X, y, percentage=0.9):
    p = int(len(X) * percentage)
    X_train = X[0:p]
    Y_train = y[0:p]
    X_train, Y_train = shuffle_in_unison(X_train, Y_train)
    X_test = X[p:]
    Y_test = y[p:]
    return X_train, X_test, Y_train, Y_test
data = pd.read_csv('AAPL.csv')[::-1]
data = data.ix[:, 'Adj Close'].tolist()
# Uncomment below to use price change time series
# data = data.ix[:, 'Adj
Close'].pct_change().dropna().tolist()
plt.plot(data)
plt.show()
WINDOW = 30
EMB_SIZE = 1
STEP = 1
FORECAST = 5
# Straightforward way for creating time windows
X, Y = [], []
for i in range(0, len(data), STEP):
    try:
        x_i = data[i:i+WINDOW]
```

```
            y_i = data[i+WINDOW+FORECAST]
            last_close = x_i[WINDOW-1]
            next_close = y_i
            if last_close < next_close:
                y_i = [1, 0]
            else:
                y_i = [0, 1]
        except Exception as e:
            print e
            break
        X.append(x_i)
        Y.append(y_i)
X = [(np.array(x) - np.mean(x)) / np.std(x) for x in X] #
comment it to remove normalization
X, Y = np.array(X), np.array(Y)
X_train, X_test, Y_train, Y_test = create_Xt_Yt(X, Y)
model = Sequential()
model.add(Dense(64, input_dim=30,
activity_regularizer=regularizers.l2(0.01)))
model.add(BatchNormalization())
model.add(LeakyReLU())
model.add(Dropout(0.5))
model.add(Dense(16,
activity_regularizer=regularizers.l2(0.01)))
model.add(BatchNormalization())
model.add(LeakyReLU())
model.add(Dense(2))
model.add(Activation('softmax'))
opt = Nadam(lr=0.001)
reduce_lr = ReduceLROnPlateau(monitor='val_acc', factor=0.9,
patience=25, min_lr=0.000001, verbose=1)
checkpointer = ModelCheckpoint(filepath="test.hdf5",
verbose=1, save_best_only=True)
model.compile(optimizer=opt,
loss='categorical_crossentropy',
metrics=['accuracy'])
```

```
history = model.fit(X_train, Y_train,
          nb_epoch = 1,
          batch_size = 128,
          verbose=1,
          validation_data=(X_test, Y_test),
callbacks=[reduce_lr, checkpointer], shuffle=True)
plt.figure()
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='best')
plt.show()
plt.figure()
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='best')
plt.show()
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
pred = model.predict(np.array(X_test))
C = confusion_matrix([np.argmax(y) for y in Y_test],
[np.argmax(y) for y in pred])
print C / C.astype(np.float).sum(axis=)
FROM = 0
TO = FROM + 500
original = Y_test[FROM:TO]
predicted = pred[FROM:TO]
plt.plot(original, color='black', label = 'Original data')
plt.plot(predicted, color='blue', label = 'Predicted data')
      plt.show()
```

ДОДАТОК Б

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	67 с.
	Інші документи	
2.	Презентаційні матеріали	18 плакатів

-	-	Прізвище та ініціали.	Підп.	Дата		
Розробив		Троценко О. І.			Шифр групи	Код напр./спец.
Перевірив		Рябова Н. В.			СШІм-20-2	122
Н.контр.		Малєєва І.А.			ХНУРЕ кафедра ШІ	
Затв.		Філатов В.О.				

Нейромережевий підхід до вирішення задач прогнозування цін цифрового активу