Warsaw University of Technology

FACULTY OF
MATHEMATICS AND INFORMATION SCIENCE

Project of an application based on machine learning for stock market prediction
User Documentation

Vladislav Sorokin

Dzianis Harbatsenka

Nikita Zakharov

thesis supervisor
prof. PW dr hab. inż., Jerzy Balicki

Version 1.0

Warsaw, 03.01.21

Table of contents

1. Deployment Documentation

1.1. System Requirements

Operating system:
Windows 10 (version 1709 or higher)

Processors:
Minimum: Any Intel or AMD x86-64 processor
Recommended: Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support

Disk:
Minimum: 3 GB of HDD space
Recommended: 5 GB of SSD space

RAM:
Minimum: 8 GB
Recommended: 16 GB

Graphics:
No specific graphics card is required.
GPU acceleration requires a GPU that supports CUDA 3 or newer.

1.2. Matlab

Verify that version 9.8 (R2020a) of the MATLAB Runtime is installed.

If not, you can run the MATLAB Runtime installer.

To find its location, enter

>>mcrinstaller

at the MATLAB prompt.

NOTE: You will need administrator rights to run the MATLAB Runtime installer.

Alternatively, download and install the Windows version of the MATLAB Runtime for R2020a

from the following link on the MathWorks website:

https://www.mathworks.com/products/compiler/mcr/index.html

For more information about the MATLAB Runtime and the MATLAB Runtime installer, see "Distribute Applications" in the MATLAB Compiler documentation in the MathWorks Documentation Center.

1.3. Python

It's highly recommended to use Python version 3.7 or above. Version used during implementation is 3.7.5. Basic Python prompt can be used. The link below forwards you to the page where you download Python.

https://www.python.org/downloads/

It's recommended to use PyCharm for code verification. To download PyCharm follow the link below.

https://www.jetbrains.com/pycharm/download/#section=windows

The solution uses external libraries and packages that will need to be installed. The whole list of them is presented here:

tensorflow==2.2.0

matplotlib==3.2.1

numpy==1.18.1

yahoo_fin==0.8.5

pandas==0.24.2

requests==2.23.0

PyQt5==5.15.2

scikit_learn==0.23.2

The above libraries and packages will be downloaded automatically using additional script prepared for this purpose. In details it will be explained in the next section.

Moreover, it's also highly recommended to install CUDA for the purpose of acceleration of the execution. The guide for installation can be found here:

https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html?ncid=afm-chs-44270&ranMID=44270&ranEAID=a1LgFw09t88&ranSiteID=a1LgFw09t88-y2dDTmA7YNjrVOTGkSr7yQ

1.4. R

Verify that R of version 4.0.3 or upper is installed. To get that version on your local machine, please follow the provided link below and fulfill the installation instructions:

https://cran.r-project.org/bin/windows/base/

Be cautioned to install the R into the folder, provided by the installer by default. In case if either another installation folder was chosen or another version of R is presented in the system, in the application folder navigate to cnnr folder, find out the RPath.txt file and change an existing path to the path of your R installation.

Code verification may be done via any convenient redactor and compilator for R, but RStudio is highly preferred. The following link allows to download the latest stable version of RStudio from the official website:

https://rstudio.com/products/rstudio/download/

The list of external packages with their versions used in implementation phase and needed for successful execution of the application is provided below:

- shiny (1.5.0)
- shinyjs (0.7.1)
- shinydashboard (2.0.0)
- tensorflow (2.2.0)
- keras (2.2.0.0)
- ggplot2 (3.3.2)
- stringr (1.4.0)
- glue (1.4.2)
- tfruns (1.4)
- caret (6.0-68)
- testthat (3.0.0)

The application will automatically download all the missing packages during the first program compilation.

2. Installation Instruction

In order to use the application please follow the installation instruction presented below.

1. Download the source files and place in some folder for your convenient usage. The source code can be found using the link below:
   https://github.com/NikitaPW/StockPricePredictionApplication

2. Install Python prompt. It is highly recommended that the version of Python should be 3.7 or above, in order for all included libraries to work correctly. The version used during implementation is 3.7.5. Python will allow to launch the main application and one of the approaches – LSTM. However, some additional installation is required. In the folder with source files in lstm folder, please, launch the install.bat. This will download and install all the required libraries and packages.
3. Install CUDA using the link presented in the previous chapter (optional)
4. Install Matlab as per the information presented in the previous chapter.
5. Install R as per the information presented in the previous chapter. All libraries will be downloaded automatically on the execution call.
6. Verify that all steps above are done. If so, installation is completed.

3. Technical Documentation

3.1. Application functionality

Below table 1 describes present functionality of the application.

No	Functionname	Function description
1	Choosealgorithms	User chooses one of three algorithms to get the future stock price and gets the detailed description for a specific algorithm.
2	Choosedataset manually	User given a choice to set dataset path manually, so the program can be launched offline
3	Choose period	User choose the period for fitting the model – 1 month, 3 months and one year.
4	Choose a stock market index	After launching the specific application user chooses a stock market index the data for which will be downloaded.
5	Fit model	Starting fitting model with downloaded data.
6	Test model	Starting testing model in order to see its correctness and results.
7	Predict future price	Letting the user predict the price for market index for some date in the future.
8	Check documentation	User can choose the option in application to see more thorough description of how to work with program

Table 1. Present functions

3.2. System architecture

The system is presented as a desktop application consisting of several components, which allow to utilize particular raw data modifications, model initialization and performance of following procedures like model training, validation and testing, which will produce a stock prediction in the end. Figure 1 shows a diagram with all dedicated components forming the application. The brief description of each component is provided below.

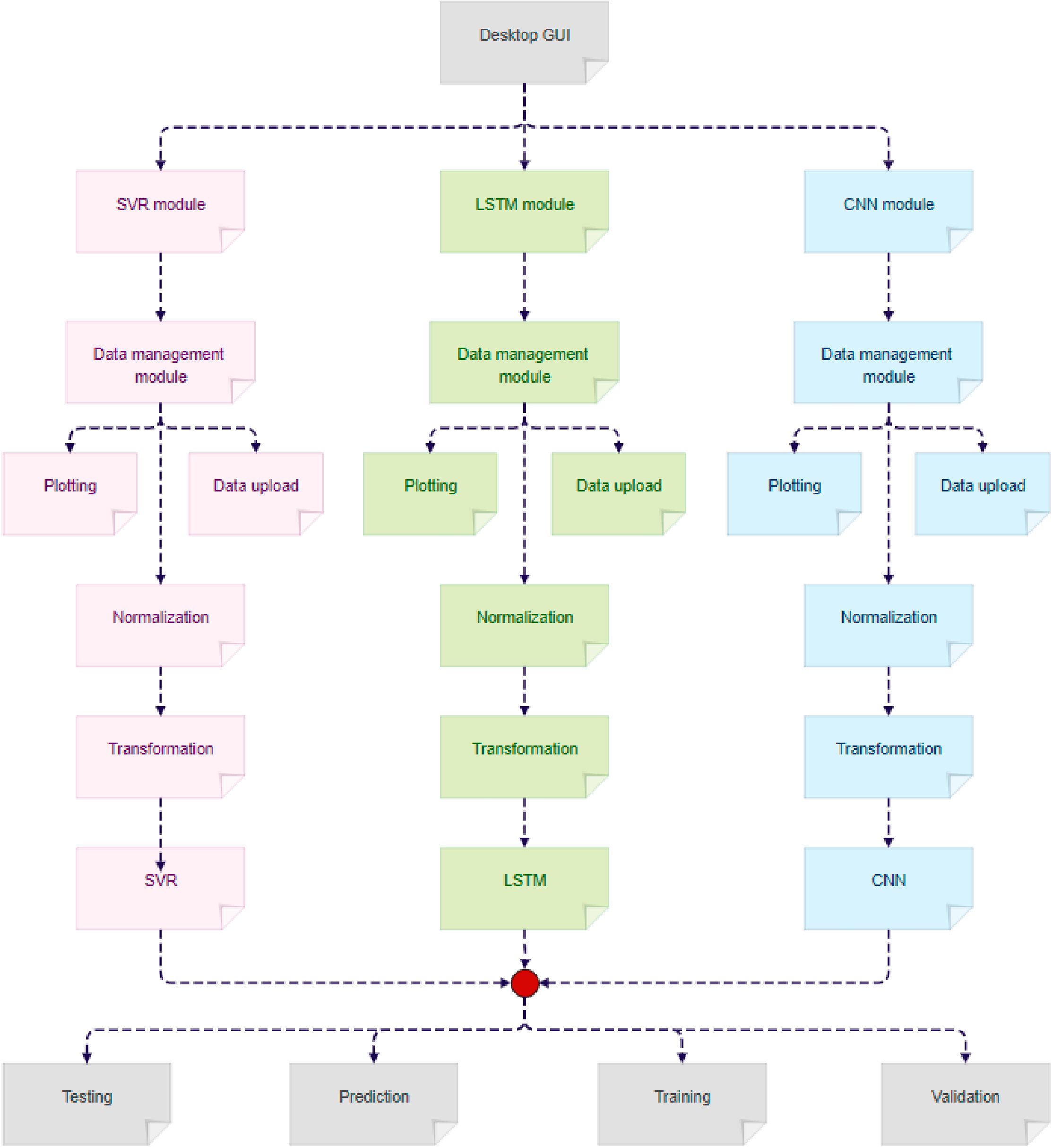


Figure 1. Component diagram

The is a list of all components used in the application with brief dedication explanation:

- **Desktop GUI** is an application interface, available for the user during initial exploitation of the program. Here user may choose, what model will be used in current session.
- **SVR module** is provided with all functionalities needed to utilize support vector machine algorithm.
- **LSTM module** consists of all components for LSTM implementation.
- **CNN module** is fully dedicated to a convolution neural network.
- • **Data management** module is a field, where particular modification of the raw input data can be performed
- • **Plotting** is the visualization of an initial raw data.

- • **Data** upload components used to download some unmodified data to the implementation.
- **Normalization** makes an input being scaled to a certain range and needed for efficient model learning.
- **Transformation** component allows to fit the normalized data into machine learning algorithm
- **SVR** – Support Vector Machine initialized with predefined hyperparameters.
- **LSTM** – Long Short-Term Memory network initialized with predefined hyperparameters.
- **CNN** – Convolution Neural Network initialized with predefined hyperparameters.
- **Training** component is dedicated to actual model training, based on chosen hyperparameters and an input.
- **Testing** is a performance test of the trained model, with certain visualization provided.
- **Validation** is used during training of a model and allow to best fit model by validation of every instance of a training.
- **Prediction** component allows to used already trained model for real stock prediction. Makes a plot of a result.

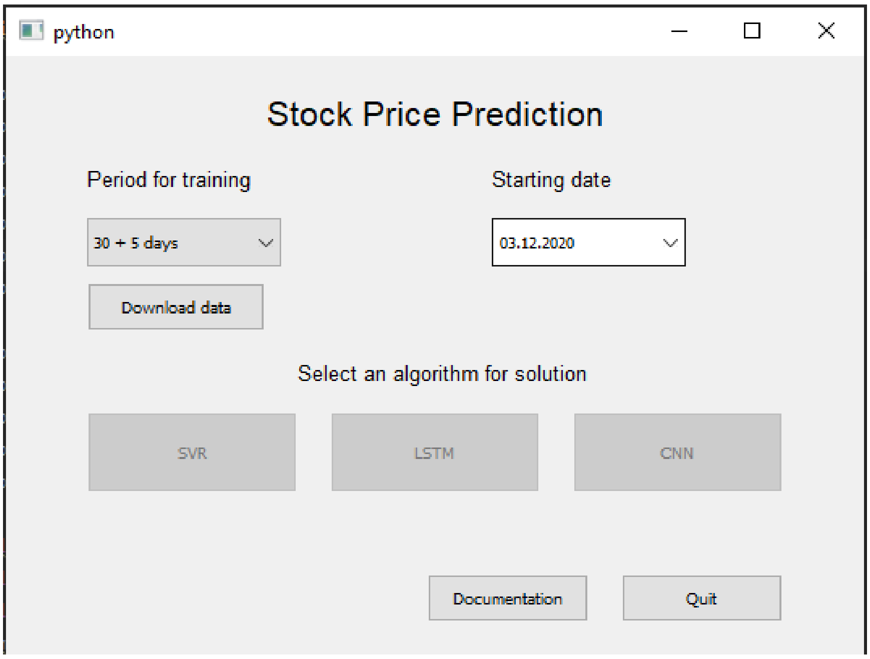
3.3. Data download

For a prediction, the index of Wig20 is taken from the website stooq.pl. We are having 3 different time periods that can be downloaded from the main program. The data is stored in CSV format, which allows to access any index needed in computations. Input data is represented in comma-separated format with several unnecessary indices.

4. User's Manual

In this section usage of the program will be presented with detailed explanation of each approach.

To launch the program, open the folder with source files and launch the runme script that will trigger the application to start. After that the following screen pops up as on Figure 2.



Period for training: 30 + 5 days

Starting date: 03.12.2020

Figure 2. Main application view.

The user is allowed to choose the data set to be downloaded to proceed for a solution. Three options are available: 1 month, 3 months and 1 year. Also user can specify the date from which the count of days will start. So if the user wants to take the period for training as first three months of 2019, then the date of 01.01.2019 should be specified and 90 + 5 days has to be chosen.

Then by clicking the download data button the corresponding data will be downloaded and then it will be used is subsequent processing.

4.1. Matlab

To start Matlab SVR solution, please click SVR button on the main screen. After successful launch of an application the following screen will be presented as on Figure 3.

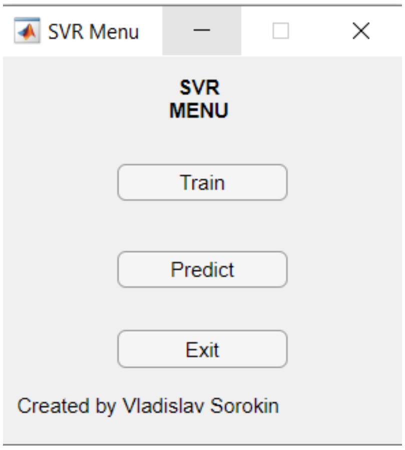
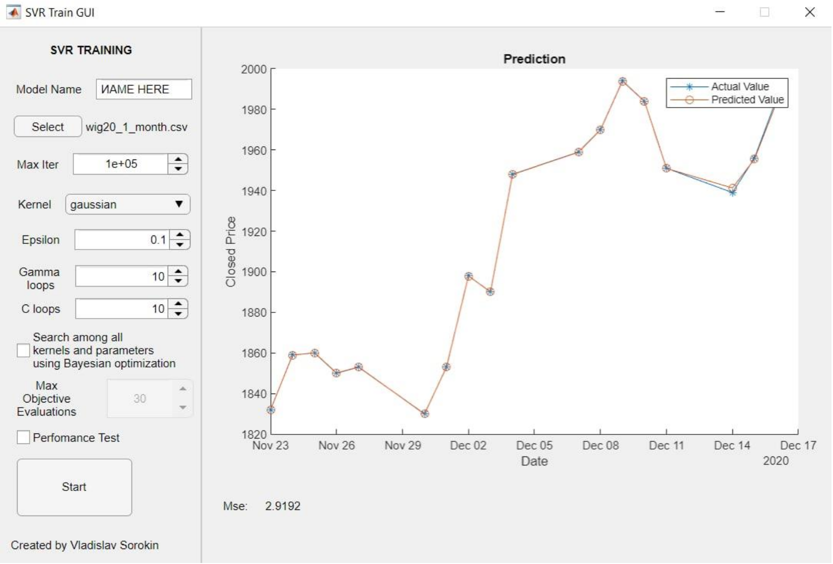


Figure 3. Matlab SVR main app view

The user is allowed to choose from three options – train model, predict future price and close the application.

In order to proceed with model train, please, click on train button and the following screen will pop up as on Figure 4.


SVR Train GUI

SVR TRAINING

Model Name ИAME HERE

Select wig20_1_month.csv

Max Iter 1e+05

Kernel gaussian ▼

Epsilon 0.1

Gamma loops 10

C loops 10

□ Search among all kernels and parameters using Bayesian optimization

Max Objective Evaluations 30

□ Perfomance Test

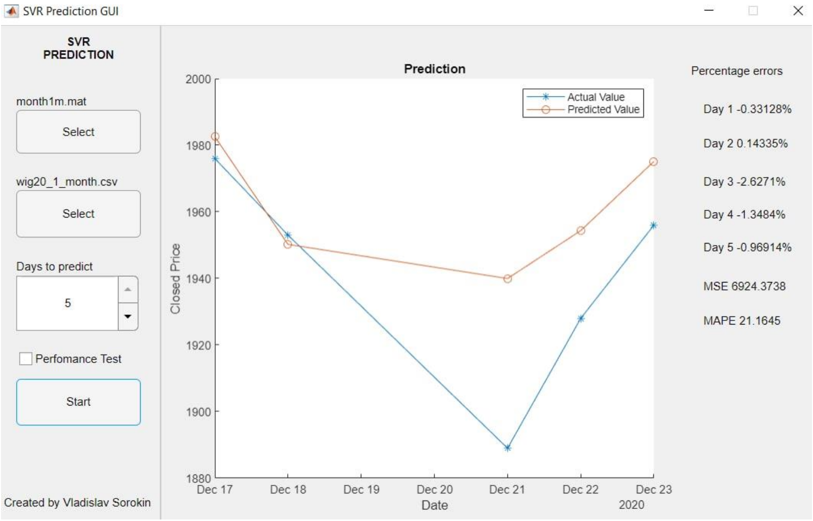
Mse: 2.9192

Figure 4. SVR train procedure

The user is given a chance to choose the model name, select data, select max of iterations that will be executed, Kernel, epsilon value, gamma loops, c loops – number of loops while tuning. Moreover, it's allowed to choose whether to use Bayesian optimization or not. Performance Test ticker tells the program whether to apply testing optimization or not.

After all parameters are set the needs to click the Start button and wait till the program finishes the execution. At this stage training procedure is finished.

Next step is to predict future values. To do that user needs to enter Predict window from main application window. It will the following as on the Figure 5.



month1m.mat

wig20_1_month.csv

Days to predict: 5

□ Perfomance Test

Percentage errors

Day 1 -0.33128%

Day 2 0.14335%

Day 3 -2.6271%

Day 4 -1.3484%

Day 5 -0.96914%

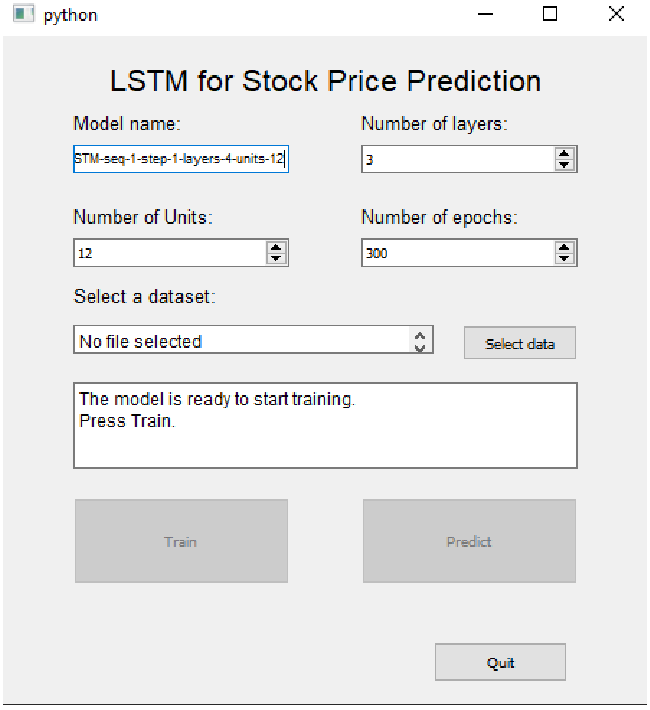
MSE 6924.3738

MAPE 21.1645

Figure 5. SVR predict phase.

4.2. Python

To start Python solution using LSTM approach from the main window the user needs to click LSTM button. This will bring the following window as on Figure 6.



LSTM for Stock Price Prediction

Model name: STM-seq-1-step-1-layers-4-units-12

Number of layers: 3

Number of Units: 12

Number of epochs: 300

Select a dataset: No file selected

The model is ready to start training.
Press Train.

Figure 6. Python LSTM main window.

The user is allowed to choose name of the model to be created, number of layers in the network, number of cells in the network, number of epochs the model will be trained. Afterwards, the data set needs to be specified. When it’s done. The user can follow a helping screen with instructions to follow. To start training the user can press Train button and the model will start training. When it’s done the following screen will show up as on Figure 7.

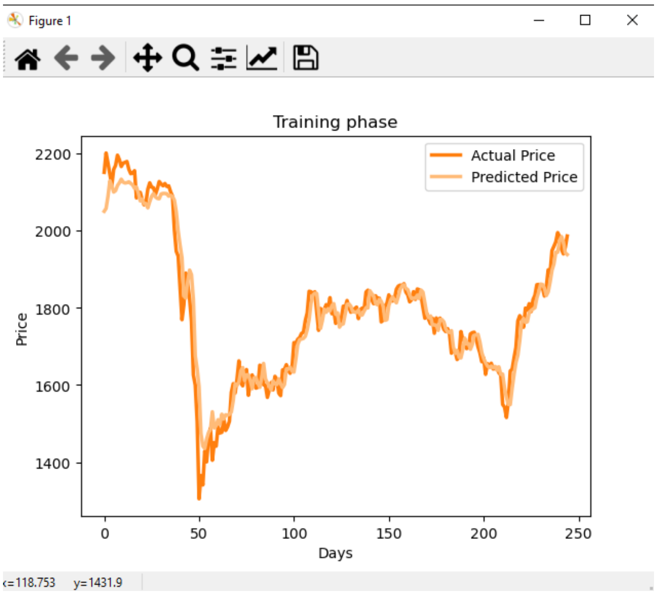


Figure 7. LSTM train process.

To start the prediction from the main window of selected approach the user can click Predict button that will start prediction phase. This will show up the graph as on the Figure 8.

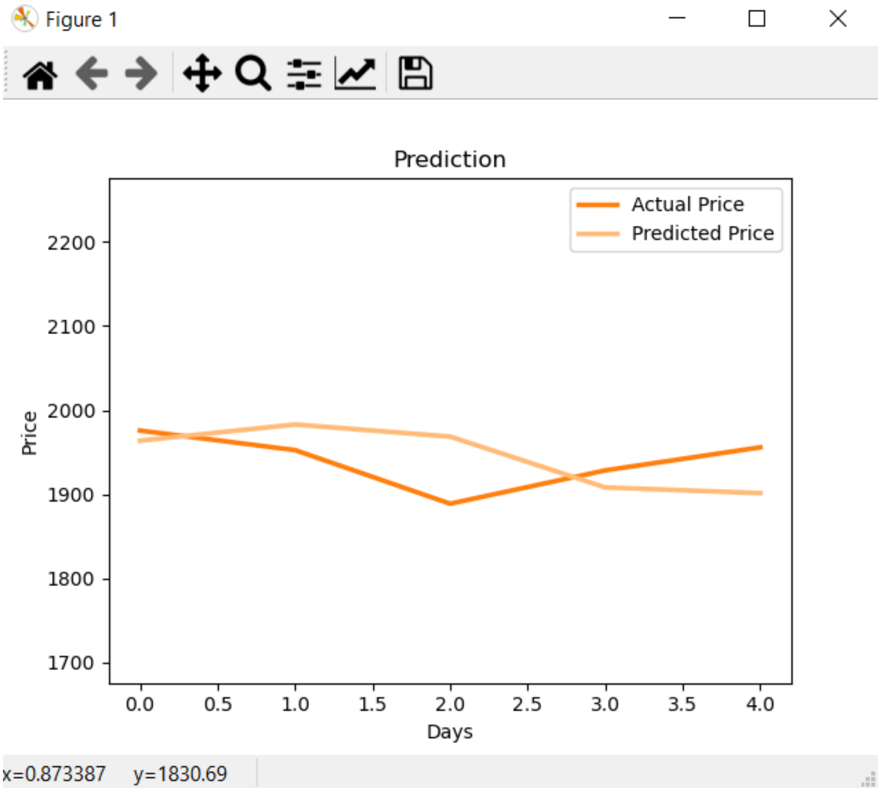
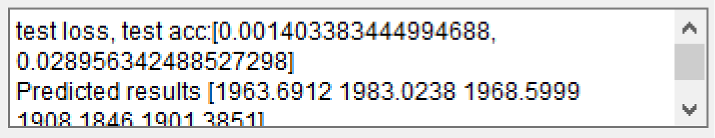


Figure 8. LSTM prediction phase.

When prediction is done the results of the predicted prices can be seen in the main window as on Figure 9. The prices are predicted for the next 5 days.

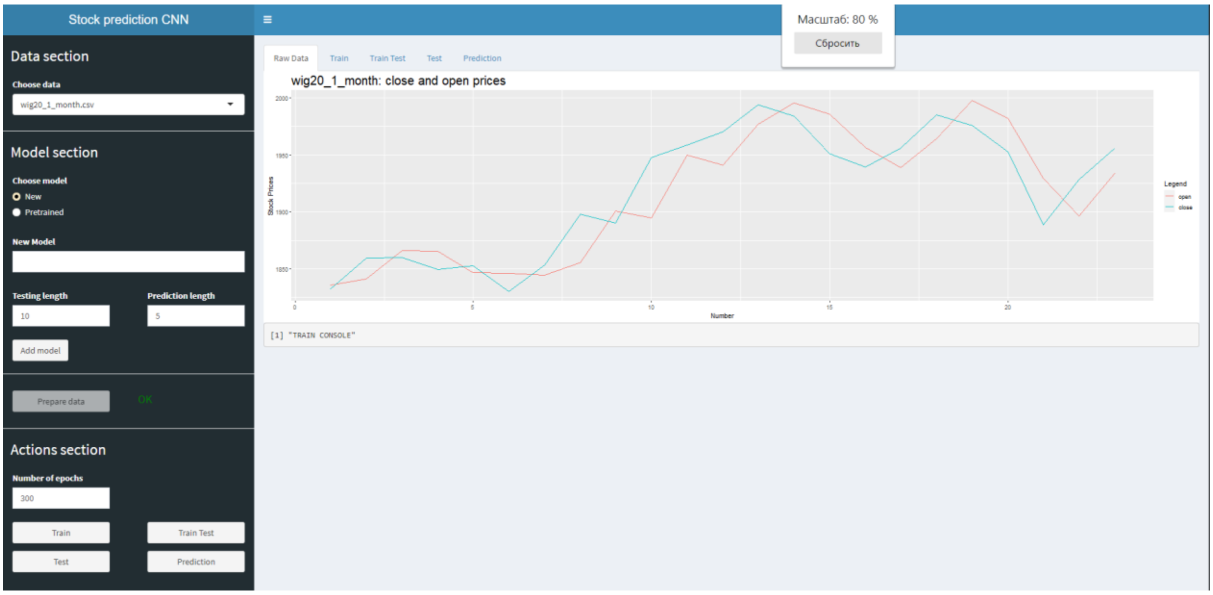


```
test loss, test acc:[0.0014033834449946888,
0.028956342488527298]
Predicted results [1963.6912 1983.0238 1968.5999
1908 1846 1901 3851]
```

Figure 9. LSTM results.

4.3. R

In order to start the R solution, the user needs to click CNN button on the main application. Then the following UI will be opened in the browser as on Figure 10.



Choose data: wig20_1_month.csv

Choose model: ○ New ● Pretrained

Testing length: 10

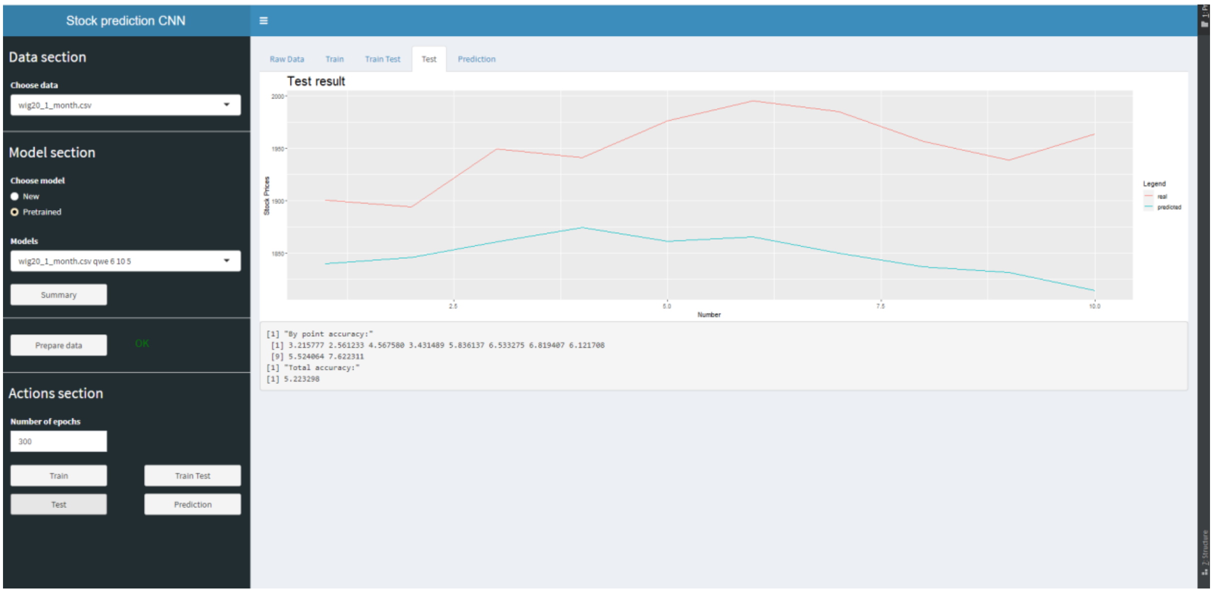
Prediction length: 5

Number of epochs: 300

```
[1] "TRAIN CONSOLE"
```

Figure 10. CNN UI

In the UI user has to choose data to be trained, whether to use already pretrained model or create the new one. Also user can change default parameters, such as Testing length and Prediction length, which define the number of days that are going to be used for testing the model and prediction. After these parameters are set, the user can proceed with training. However, please make sure that the model is chosen or added, in case of the new one. To train the model simply press the Train button. After the training is done (the user will be notified), press Test to see how model behaves on testing set. To get predicted values, simply press Prediction button. The values will pop up on the screen as it’s shown on Figure 11.



Choose data: wig20_1_month.csv

Choose model

○ New

● Pretrained

Models: wig20_1_month.csv qwe 6 10 5

Number of epochs: 300

```
[1] "By point accuracy:"
 [1] 3.215777 2.561233 4.567580 3.431489 5.836137 6.533275 6.819407 6.121708
 [9] 5.524064 7.622311
[1] "Total accuracy:"
[1] 5.223298
```

Figure 11. CNN prediction.

5. Glossary

CNN – Convolution neural network
CUDA - Compute Unified Device Architecture
GB – Gigabyte
GPU - Graphics processing unit
HDD – Hard disk drive
LSTM - Long short-term memory
SVR - Support Vector Regression
RAM - Random-access memory
SSD - Solid-state drive